

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

دعای پیش از مطالعه

اَللّٰهُمَّ اَخْرِجْنِيْ مِنْ ظُلُمَاتِ الْوَهْمِ وَاكْرِمْنِيْ بِنُوْرِ الْفَهْمِ
اَللّٰهُمَّ افْتَحْ عَلَيْنَا اَبْوَابَ رَحْمَتِكَ وَاَنْشُرْ عَلَيْنَا خَزَائِنَ عُلُوْمِكَ
بِرَحْمَتِكَ يَا اَرْحَمَ الرَّاحِمِيْنَ

پروردگارا، خارج کن مرا از تاریکی های فکر و گرامی بدار به نور فهم
پروردگارا، بگشای بر ما در های رحمت را و بگستران کنج های دانشت را به امید رحمت
تو ای مهربان ترین مهربانان

بیایید به حقوق دیگران احترام بگذاریم

دوست عزیز، این کتاب حاصل دسترنج چندین ساله ی مؤلف، مترجم و ناشر آن است. تکثیر و فروش آن به هر شکلی بدون اجازه از پدیدآورنده کاری غیراخلاقی، غیرقانونی، غیرشرعی و کسب درآمد از دسترنج دیگران است، نتیجه ی این عمل نادرست، موجب رواج بی اعتمادی در جامعه و بروز پی آمدهای ناگوار در زندگی و محیط ناسالم برای خود و فرزندانمان می گردد.



درسنامه اصول برنامه‌نویسی و ساختمان داده‌ها

ویژه کارشناسی ارشد رشته‌ی انفورماتیک پزشکی

**تألیف و گردآوری:
سلیمان سعیدی**

سرشناسه

عنوان و نام پدیدآور

مشخصات نشر

مشخصات ظاهری

شابک

وضعیت فهرست‌نویسی

موضوع

: سعیدی، سلیمان، ۱۳۶۸-

: AGK درسنامه اصول برنامه‌نویسی و ساختمان داده‌ها: ویژه کارشناسی ارشد رشته‌ی انفورماتیک پزشکی/

: تألیف و گردآوری سلیمان سعیدی.

: تهران: آناهید گستر خلیلی، ۱۴۰۳.

: ۲۹۰ص؛ ۲۹×۲۲ س.م.

: 978-622-4907-31-8

: فیپا

: برنامه‌نویسی -- آزمون‌ها و تمرین‌ها (عالی)

Computer programming -- Examinations, questions, etc. (Higher)

: برنامه‌نویسی -- مسائل، تمرین‌ها و غیره (عالی)

Computer programming -- Problems, exercises, etc. (Higher)

: ساختار داده‌ها -- آزمون‌ها (عالی)

Data structures (Computer science) -- Examinations (Higher)

: ساختار داده‌ها -- مسائل، تمرین‌ها و غیره (عالی)

Data structures (Computer science) -- Problems, exercises, questions, etc. (Higher)

: QA۷۶/۷۶

: رده‌بندی کنگره

: ۰۰۵/۱۳۳

: رده‌بندی دیویی

: ۹۸۲۵۱۸۰

: شماره کتابشناسی ملی



انتشارات آناهید گستر خلیلی

نام: AGK درسنامه اصول برنامه‌نویسی و ساختمان داده‌ها

(ویژه کارشناسی ارشد رشته‌ی انفورماتیک پزشکی)

تألیف و گردآوری: سلیمان سعیدی

ناشر: آناهید گستر خلیلی

نوبت و سال چاپ: اول. ۱۴۰۳

شمارگان: ۱۰۰۰

چاپ و صحافی: گیلان

مدیر تولید: امیرحسین خلیلی

مدیر فنی و هنری: مریم آرده

تایپ و صفحه‌آرایی: آرزو راضی

بهاء: ۳۳۰۰۰۰ تومان

آموزشگاه/ فروشگاه: ۰۲۱۶۶۵۶۸۶۲۱

مرکز پخش: ضلع جنوب غربی میدان انقلاب . جنب بانک پارسیان . مجتمع تجاری پارس . طبقه اول

مدیر فروش: ۰۹۱۹۱۶۸۵۹۷۰ (ارتباط از طریق پیامک)

 agkpublisher@gmail.com

 www.agkins.com

طلیحه سخن مدیر تولید:

سپردن کار به اهلش نشانه‌ی خردمندی است.

برای تولید بسته‌های آموزشی تمام تلاش خود را به کار گرفته‌ایم تا محصولی شایسته را تقدیم شما عزیزان نماییم.

مجموعه‌ای که پیش روی شماست براساس معتبرترین و به‌روزترین منابع کنکور و به قلم مجرب‌ترین اساتید تألیف شده است و مطالعه آن موجب صرفه‌جویی در زمان و هزینه شما خواهد شد. ارائه بالاترین آمار قبولی در کنکور کارشناسی ارشد و دکتری در دهه اخیر بیان‌گر توانایی این مجموعه در راستای اهداف آموزشی خود می‌باشد. اینک با اتکا به تجربه‌ای ۱۷ ساله می‌توانیم ادعا کنیم که مجموعه حاضر، بی‌شک از جمله کامل‌ترین و به‌روزترین بسته‌های آموزشی موجود در عرصه کنکور می‌باشد.

❧ برخی از ویژگی‌های بسته‌های آموزشی:

- منطبق با معتبرترین و به‌روزترین منابع کنکور
- گردآوری و تألیف توسط مجرب‌ترین اساتید کنکور
- تدریس مطالب از سطح پایه تا پیشرفته به بیان ساده و روان
- انتقال بهتر مفاهیم با استفاده از شکل، جدول و نمودار
- تألیف مبتنی بر نحوه‌ی طراحی سوالات کنکور
- مطالعه یک منبع منسجم به‌جای مطالعه چندین منبع مختلف
- منبعی خودآموز و بدون نیاز به مدرس

اینک علاوه بر مطالعه بسته‌های آموزشی استفاده از خدمات و محصولات زیر را به عنوان ضمانتی برای قبولی به شما داوطلبان عزیز توصیه می‌کنم؛

❧ آزمون‌های آزمایشی: خوش بود گر محک تجربه آید به میان

داوطلب شجاع کیست؟ داوطلبی است که خود را مورد سنجش (خود را با خود) و ارزیابی (خود را با دیگران) قرار دهد تا پیش از کنکور به نقاط ضعف خود پی ببرد و آن‌ها را تقویت نماید. آزمون‌های آزمایشی دارای یک برنامه‌ریزی منسجم است و به شما اولتیماتوم می‌دهد. یعنی شما باید در یک زمان معین مطالب معینی را مطالعه نمایید.

آزمون‌های آزمایشی مجموعه دارای سوالات استاندارد، پاسخنامه تشریحی و کارنامه تحلیلی می‌باشد. صادقانه شرکت در آزمون‌های آزمایشی را به شما عزیزان توصیه می‌کنم.

❧ مشاوره تخصصی و برنامه‌ریزی: زیر نظر حرفه‌ای‌ها درس بخوانید ...

در صورتی که نمی‌توانید با توجه به زمان آزادتان برنامه‌ی مطالعاتی استاندارد را برای خود تدوین کنید حتماً از افراد متخصص کمک بخواهید. افرادی که تجربیات خود را صادقانه در اختیار شما قرار می‌دهند و برنامه‌ریزی مناسب با شرایط شما را انجام می‌دهند.

📖 کتاب تست:

کوچک‌ترین واحد یادگیری در کنکور، تست است.

شما می‌توانید برای این‌که سطح یادگیری خود را پس از مطالعه بسنجید از کتاب‌های تست مجموعه (تست کنکور) و ماطراحان (تست‌های تألیفی) استفاده کنید. این کتاب‌ها به‌صورت موضوعی طبقه‌بندی شده‌اند و می‌توانند نحوه‌ی طرح سوالات کنکور و سطح آن‌ها را به‌صورت واضح به شما نشان دهند.

📖 درسنامه‌های نکات داغ و کتاب‌های الگوریتم مناسب‌ترین منابع برای دوران مرور و جمع‌بندی:

بیش‌تر داوطلبان حین مطالعه نگران خلاصه‌نویسی و مرور آموخته‌های خود هستند و برای خلاصه‌نویسی زمان زیادی را صرف می‌کنند. اینک به شما مژده می‌دهیم که این‌بار زمان به نفع شما در حال گذر است و دیگر با استفاده از درسنامه‌های نکات داغ و کتاب‌های الگوریتم نیاز به خلاصه‌نویسی ندارید.

شما می‌توانید نظرات، انتقادات و پیشنهادهای سازنده خود را در مورد بسته‌های آموزشی به شماره‌تلفن همراه ۰۹۱۹۱۶۸۵۹۷۰ فقط و فقط پیامک نمایید.

مشاور و مدیر تولید

امیرحسین خلیلی

طلیحه سخن مؤلف:

حمد و سپاس بی‌کران به درگاه ایزد یکتا که توفیق تهیه‌ی این درسنامه را عطا فرمود.

نگاهی اجمالی به تعداد داوطلبان آزمون کارشناسی ارشد نشان می‌دهد که در این سال‌ها درخواست برای تحصیل در دوره‌های تحصیلات تکمیلی دانشگاه‌ها رشد چشمگیری داشته است. دشواری پیش روی بیش‌تر داوطلبان، گوناگونی منابع درسی و عدم دسترسی به آن‌ها می‌باشد. نکته قابل توجه تسهیل دسترسی داوطلبان به کتاب‌هایی است که شامل خلاصه‌ها و نکات جامع و مهم از منابع است که مورد نظر طراحان سوال می‌باشد. لذا با توجه به این مهم، مؤلف تلاش کرده است تا کامل‌ترین درسنامه را به داوطلبان ارائه دهد. هدف اصلی این درسنامه ارائه نکات مهم و کنکوری درس اصول برنامه‌نویسی پاسکال و C (سی) و ساختمان داده‌ها جهت کمک به موفقیت داوطلبان آزمون کارشناسی ارشد رشته انفورماتیک پزشکی می‌باشد. درسنامه حاضر شامل سه بخش بوده که بخش اول؛ خلاصه‌ها و نکات مهم و کنکوری زبان برنامه‌نویسی پاسکال و بخش دوم؛ خلاصه‌ها و نکات مهم و کنکوری زبان برنامه‌نویسی C (سی) و بخش سوم؛ نکات مهم و کنکوری ساختمان داده‌ها می‌باشد. برای تهیه این مجموعه از کتاب‌ها و منابع معتبری همچون کتاب کامل و بسیار عالی درس و کنکور پاسکال، درس و کنکور C (سی) و درس و کنکور ساختمان داده‌ها تالیف مهندس حمیدرضا مقسمی و هم‌چنین کتاب‌های بسیار عالی درس و کنکور C و ++C و درس و کنکور ساختمان داده‌ها تالیف مهندس فرشید شیرافکن استفاده شده است. لذا از این عزیزان نهایت سپاس و قدردانی را داریم.

در پایان از انتشارات آناهید گستر خلیلی که در تهیه این درسنامه با بنده همکاری نموده‌اند، تقدیر و تشکر می‌نمایم.

با احترام
موفقیت همگان، آرزوی ماست.
سلیمان سعیدی

بخش اول: پاسکال

فصل اول: ساختار برنامه‌های پاسکال	۱۰
فصل دوم: متغیرها و ثابت‌ها	۱۶
فصل سوم: عملگرها و عبارات	۲۸
فصل چهارم: دستورات ورودی و خروجی	۳۷
فصل پنجم: دستورات کنترلی	۴۶
فصل ششم: آرایه و رشته	۵۷
فصل هفتم: زیربرنامه‌ها (تابع و پروسجر)	۶۹
فصل هشتم: گونه شمارشی، مجموعه و رکورد	۸۵
فصل نهم: فایل‌ها در پاسکال	۹۸
فصل دهم: اشاره‌گرها	۱۱۱

بخش دوم: زبان C

فصل اول: ساختار برنامه C	۱۲۱
فصل دوم: متغیرها و ثابت‌ها	۱۲۷
فصل سوم: عملگرها و عبارات	۱۳۵
فصل چهارم: سیستم‌های ورودی و خروجی	۱۴۳
فصل پنجم: دستورات کنترلی	۱۵۲
فصل ششم: آرایه و رشته	۱۶۳
فصل هفتم: اشاره‌گرها	۱۷۰
فصل هشتم: توابع	۱۸۱
فصل نهم: نوع شمارشی، استراکچر و یونیون	۱۹۳
فصل دهم: فایل	۲۰۱

بخش سوم: ساختمان داده‌ها

فصل اول: الگوریتم و پیچیدگی زمانی (اجرایی)	۲۰۹
فصل دوم: آرایه‌ها	۲۱۹
فصل سوم: پشته	۲۲۷
فصل چهارم: صف	۲۳۰
فصل پنجم: لیست پیوندی	۲۳۹
فصل ششم: درخت‌ها	۲۵۰
فصل هفتم: گراف	۲۷۲
فصل هشتم: مرتب‌سازی	۲۸۲

بخش اول

پاسکال

فصل

۱

ساختار برنامه‌های پاسکال

پاسکال ۱۰

- * اولین زبان سطح بالا (ساخت یافته) به نام Algol در سال ۱۹۶۰ ایجاد شد.
- * Algol برای الگوریتم‌های ریاضی مناسب بود و کمبودهایی از جمله فاقد نوع‌های مهم کارکتر و اشاره‌گر بود داشت.
- * ۱۹۷۰ ← ایجاد زبان پاسکال توسط نیکلاس ویرث.
- پاسکال: پدید آوردن زبانی ساخت‌یافته و خوب جهت آموزش برنامه‌نویسی و نوشتن انواع برنامه‌های علمی و تجاری.

*** ویژگی‌های زبان پاسکال**

- * ساخت‌یافته (سطح بالا) یعنی می‌توان آن را به بلوک‌ها و قطعاتی کوچک‌تر تقسیم کرد تا برنامه زود کامپایل شود.
- * در زمینه‌ی آموزشی - گرافیکی - بانک اطلاعاتی - تجاری - حسابداری و ... استفاده می‌گردد (یک زبان همه منظوره (general purpose))
- * در نوشتن برنامه پاسکال از مفهوم بلوک استفاده می‌شود هر بلوک ما بین دو علامت begin (شروع بلوک یا برنامه اصلی) و End (پایان بلوک یا برنامه اصلی) قرار گرفته و در حکم یک جمله می‌باشد.
- * از دستورات go to (دستوری جهت پرش به برچسب یا Label ایجاد شده در برنامه) حتی الامکان استفاده نمی‌شود.
- * دارای سه نوع خطا: ۱- Compiletime errors (خطای Syntax) که خطای گرامری دستورات بوده اگر برنامه این خطا را داشته باشد تبدیل به کد ماشین نمی‌شود. ۲- خطای زمان اجرا (run-time errors) مانند زمان اجرا که عددی تقسیم بر صفر ($\frac{x}{0}$) شود.
- ۳- خطای منطقی (Logical error) در هنگام اجرا یا کامپایل برنامه پیامی صادر نمی‌شود ولی جوابی از برنامه می‌گیریم که انتظار آن را نداریم ← مثال: نسبت دادن متغیر صحیح دو بایتی به متغیر صحیح یک بایتی.
- Modula-2, Ada, C, Pascal جزو زبان‌های ساخت‌یافته و BASIC, FORTRAN و COBaL جزو غیر ساخت یافته (Non-Structural) هستند.
- نوشتن و اشکال‌یابی (Debugging) برنامه و توسعه آینده ساخت یافته آسان‌تر است و خواندن و درک برنامه راحت است به دلیل این که به اجزای کوچک‌تر تبدیل می‌شود.

*** ساختار زبان پاسکال**

- * در زبان پاسکال حروف کوچک و بزرگ با هم فرقی ندارند. (VAR معادل var ...)
- ۱- در خط اول برنامه اغلب نام اختیاری برنامه سپس کلمه کلیدی Program آورده می‌شود. که عنوان یا هدر (Header) برنامه نام دارد. (عنوان برنامه در پاسکال استاندارد اجباری ولی در توربو پاسکال اختیاری است). ← Program q_۳ (Input , output);

✚ انتهای خطوط برنامه باید علامت ; نوشته شود.

✚ بعد از Begin سمی کالن (;) نباید نوشته شود و همچنین بعد از END آخر برنامه به جای علامت نقطه (END.) باید گذاشت.

✚ نوشتن; برای جمله قبل از هر END اختیاری است.

✚ توضیحات (Remark یا Gomment) در زبان پاسکال داخل علامت { } یا علامت (* *) نوشته می‌شوند. و این توضیحات فقط برای نشان دادن هدف آن دستور یا برنامه به خواننده استفاده می‌شود و توسط کامپایلر نادیده گرفته می‌شوند.

✚ متغیرهای بعد از کلمه کلیدی VAR در ابتدای برنامه اصلی (متغیرهای سراسری) یا بلوک‌ها (متغیرهای محلی) تعریف می‌شوند و برای تعریف متغیرها ابتدا نام متغیر و سپس نام دیتا تایپ (Data type) آورده می‌شود: Var I, C: Real * در یک خط می‌توان چند متغیر از یک نوع را همزمان تعریف کرد. (برخلاف ثابت‌ها)

✚ ۲- از دستورات Write و Writeln برای نمایش پیام یا محتوای متغیرها بر روی مونیتر استفاده می‌شود و پیام‌ها داخل علامت تک‌گیومه نوشته می‌شوند و هر آنچه داخل تک‌گیومه باشد عیناً روی مونیتر چاپ می‌شود که دستور Write پیام را نوشته و مکان‌نما جلو پیام در همان خط باقی می‌ماند ولی دستور Writeln پس از چاپ پیام مکان‌نما را به خط بعد می‌برد: اگر نام متغیر جلوی Writeln بیاید محتوای آن چاپ می‌شود:

$$\begin{cases} \text{Write('a=?')} \\ \text{Write('bia')} \end{cases} \Rightarrow a=? \text{bia}$$

$$\begin{cases} \text{WriteLn('a=?')} \\ \text{WriteLn('bia')} \end{cases} \rightarrow \begin{cases} a=? \\ \text{bia} \end{cases}$$

$$a=6, \text{WriteLn('b=?', a)} \Rightarrow b=6$$

✚ ۳- برای خواندن ورودی‌های از کی‌بورد از دستور Read یا Readln استفاده می‌شود $\leftarrow \text{Read (n)}$ عدد n را از ورودی خوانده و آن را در برنامه قرار می‌دهد و در n ذخیره می‌شود.

✚ علامت جایگزینی در زبان پاسکال = می‌باشد که متغیر سمت راست را در متغیر سمت چپ می‌ریزد: $\leftarrow$ محتویان b در a ریخته می‌شود $a:=b$ طرز نوشتن خطوط برنامه کاملاً آزاد و اختیار است و شرط لازم برای کامپایلر این است که خطوط به ترتیب و پشت سر هم قرار گرفته باشند.

✚ دستور Clrscr () در برنامه صفحه مانیتور را پاک می‌کند و تعریف این دستور در کتابخانه (یونیت) crt قرار دارد.

✚ دستور Uses برای استفاده از unit‌های پاسکال که برنامه‌های از پیش نوشته شده می‌باشند به کار می‌رود.

✚ ۴- به کمک بخش Const (Constant (ثابت)) می‌توان ثابت‌هایی را تعریف کرد و یک ثابت نامی مجازی است که به یک عدد یا رشته داده می‌شود. مثلاً $P=3.5 \leftarrow$ نام مجازی P را می‌دهد از این به بعد هنگام کامپایل برنامه هر جا کامپایلر به اسم P برسد آن را پاک کرده و به جای آن عدد 3.5 را قرار می‌دهد.

✚ ثابت برخلاف متغیرها فضایی در حافظه اشغال نمی‌کند و همچنین ثابت‌ها در برنامه قابل مقداردهی نیستند یعنی اگر $P:=3.5$ می‌بود خطای کامپایلری رخ می‌دهد.

✚ به‌طور کلی برنامه‌های بلوک‌های پاسکال از دو قسمت ۱- تعاریف یا اعلانات (مانند: Label, type, ثابت Const و متغیرها Var و ...).

✚ ۲- دستوری که بخش دستورات در بین دو علامت begin و End قرار می‌گیرند.

✚ مراحل تبدیل یک فایل Source (منبع) پاسکال (که اغلب پسوند pas دارد) به یک فایل اجرایی (با پیوند exe) به صورت زیر است:

(.exe) اجرایی $\rightarrow$ (.obj) مقصد $\rightarrow$ (.pas) منبع

✚ هر برنامه پاسکال از سه قسمت ۱- عنوان برنامه، ۲- تعاریف برنامه و ۳- بلوک‌های برنامه تشکیل شده است.

✚ در یونیت Graph ثابت‌ها و دستورات گرافیکی و در یونیت Dos دستورات مربوط به فایل‌ها و فهرست‌ها تعریف شده است و توربو پاسکال ۷ حدود ۲۰ یونیت را عرضه کرده است.

فصل

۲

متغیرها و ثابت‌ها

* شناسه‌ها و کلمات کلید Identifiers And Key Words

- ۱- شناسه‌ها نام‌هایی است که به توابع، متغیرها داده می‌شود.
- ۲- توربو پاسکال حداکثر ۶۳ کارکتر ابتدای یک شناسه را در نظر می‌گیرد ولی در بعضی مترجم‌های دیگر پاسکال فقط ۸ حرف اول نام‌ها در نظر گرفته می‌شود.
- اولین کارکتر باید یک حرف یا زیر خط (b-) باشد و کارکتر بعد حروف، اعداد یا زیر خط باشد.
- SHAPOL, Farid, SH-LAL, test5t
- در نام شناسه نباید جای خالی (blank) و علائم مثل: ؟، !، &، ... را به کار برد و همچنین از علامت زیر خط می‌توان استفاده کرد ولی علامت منهی را نمی‌توان استفاده کرد در صورت استفاده غلط می‌باشند.
- نام شناسه‌ها نمی‌تواند از کلمات کلیدی باشد (به کلمات کلیدی ذخیره شده (Reserved Words) می‌گویند
- مثال: «در اغلب کتاب‌ها کلمات ذخیره شده را با حروف بزرگ می‌نویسند.»
- AND , OR , THEN , REPEAT , OF , TYPE , SHL , LIBRARY... و
- نام دو متغیر یا یک متغیر و یک زیر برنامه (تابع و پروسجر) در یک بلوک نمی‌تواند یکسان باشد ولی در دو بلوک مختلف می‌تواند یکسان باشد.

انواع شناسه‌ها

- ۱- استاندارد: کلماتی هستند که توسط کامپایلر برای منظور خاص استفاده شده‌اند. شناسه‌های استاندارد را می‌توان دو بار تعریف و استفاده کرد برخلاف کلمات ذخیره شده.
- ۲- غیر استاندارد: نه جزو کلمات ذخیره شده‌اند و نه جزو شناسه‌های استاندارد. جهت نام متغیرها، توابع، برچسب‌ها و ... استفاده می‌شوند.
- مثال برای شناسه‌های استاندارد: abs , boolean , eof , input , intear , read , write , ... و
- در یک برنامه هنگامی که برنامه‌نویس شناسه‌های استاندارد را مجدداً تعریف می‌کند طبق تعریف جدید عمل می‌کند.
- از دیتا تایپ‌های مختلف فقط String کلمه کلیدست بقیه مثل Real , integer و ... شناسه استاندارد هستند.

* Data type (انواع داده‌ها)

Var

متغیرهای از نوع صحیح می‌باشند. a , b ← integer;

متغیر از نوع اعشاری می‌باشد. c ← Real;

انواع داده در پاسکال استاندارد و محدوده و فضایی که در حافظه اشغال می‌کنند:

- ۱- دیتا تایپ Char ← مقدار فضایی که اشغال می‌کند ۱ بایت می‌باشد و هر یک از ۲۵۶ کارکتر جدول کداسکی را ذخیره می‌کند.
 - ۲- دیتا تایپ Integer ← ۲ بایت حافظه اشغال می‌کند و اعداد بین ۳۲۷۶۸- تا ۳۲۷۶۷+ را ذخیره می‌کند.
 - ۳- دیتا تایپ Boolean ← ۱ بایت حافظه اشغال کرده و مقادیر False (صفر) و True (غیرصفر) را ذخیره می‌کند.
 - ۴- دیتا تایپ Real ← ۶ بایت حافظه اشغال کرده و اعداد اعشاری با ۱۱ تا ۱۲ رقم را ذخیره می‌کند.
- پاسکال استاندارد نوع String ندارد.
- تعدادی از دیتا تایپ‌های توربو پاسکال:

Shortint ← ۱۲۸- تا ۱۲۷+ ← ۱ بایت
 Byte ← ۲۵۵ (بدون علامت). ← ۱ بایت
 Logint ← ۲- میلیارد تا ۲+ میلیارد. ← ۴ بایت
 Word ← ۶۵۵۳۵ (بدون علامت) ← ۲ بایت
 Comp, Double ← اعداد اعشاری خیلی بزرگ ← ۸ بایت
 Single ← اعشاری با ۷ تا ۸ رقم ← ۴ بایت
 String ← رشته‌ای با حداکثر ۲۵۵ کارکتر ← حداکثر ۲۵۶ بایت

اگر طول رشته ۲۰ کارکتر باشد ۲۱ بایت حافظه اشغال می‌کند.

دیتا تایپ‌های ۸۰۸۷ ← Comp, extended, Double, Single.

انواع دیتا تایپ‌ها در توربو پاسکال: صحیح- اعشاری- نوع ۸۰۸۷.

* انواع داده ساده در پاسکال

اردینال:

یعنی عدد یا کارکتر یا شماره قبل و بعد از آن معلوم است به قولی گسسته هستند.

مثال: shortint, byte, word, longint, char, boolean, Integer

غیر اردینال:

یعنی عدد قبل یا بعد از آن معلوم نیست به قولی پیوسته هستند:

مثال: String- Real

به نوع اردینال نوع قابل شمارش یا اسکالر (Scalar) گویند و گونه‌های شمارش و زیر قلمروها نیز اردینال هستند.

* مقادیر ثابت

۱- عددی (۱): صحیح (integer) (ب) اعشاری (Real و ...)

۲- کارکتری (Char)

۳- رشته‌ای (String)

۴- بولین (boolean)

* مقادیر ثابت عددی

مقادیر صحیح (integer):

۱- مقادیر صحیح در پاسکال در مبنای ۱۰ (دسیمال) و مبنای ۱۶ (هگزادسیمال) نمایش داده می‌شوند.

۲- داخل اعداد صحیح نباید نقطه به کار برد.

قبل اعداد مبنای ۱۶ علامت \$ استفاده می‌شود و طول اعداد مبنای ۱۶ نمی‌تواند بیش‌تر از ۸ رقم باشد و محدوده‌ی اعداد

\$00000000 تا \$FFFFFFFF می‌باشد و در پاسکال اعداد مبنای ۱۶ فقط مثبت می‌باشند (برخلاف توربو پاسکال).

۱۰- → (-\$A) Write در توربو پاسکال درست است.

* مقادیر اعشاری

۱- به دو صورت نماد علمی و ممیز شناور نوشته می‌شوند.

فصل

۳

عملگرها و عبارات

۲۸

➤ اپراتور (عملگر) جایگزین در پاسکال = می باشد که متغیر سمت راست را در متغیر سمت چپ می ریزد و سمت چپ فقط می تواند نام یک متغیر باشد.

➤ سمت راست و چپ عملگر جایگزینی باید با هم سازگار باشند.

➤ ۱- تمام انواع داده های صحیح مانند: Shortint, Logint, Word, byte, Integer با هم سازگار می باشند.

➤ اگر ظرفیت سمت راست از ظرفیت سمت چپ بیش تر باشد خطای منطقی رخ می دهد.

➤ اگر عددی بزرگ تر از محدوده ی متغیر را بخواهیم مستقیماً در آن بریزیم خطای کامپایلری رخ می دهد ولی اگر آن عدد را ابتدا در ظرفی بزرگ تر ریخته و سپس در آن بریزیم خطای منطقی رخ می دهد:

```
Var
i: integer;
b: byte;
begin
i:=120;
b:=i;
writeln (b); → 120
i:=258;
b:=i;
writeln (b); → 2 (ظرفیت b کم تر از i خطای منطقی)
b:=257; → (ظرفیت b کم تر) خطای کامپایلری
End.
```

➤ ۲- تمام انواع اعشاری مانند Comp, Single, Extended, double, Real با هم سازگار می باشند.

➤ متغیر Real با ۶ بایت می تواند اعداد $2/9 \times 10^{-39}$ تا $1/7 \times 10^{38}$ را با ۱۱ تا ۱۲ رقم دقت در خود ذخیره کند و نوع double، ۱۵ تا ۱۶ و نوع Single، ۷ تا ۸ رقم، نوع Extended ۱۹ تا ۲۰ رقم دقت دارند.

➤ نوع Comp اعشاری می باشد و برای ذخیره اعداد صحیح خیلی بزرگ (۱۹ تا ۲۰ رقم دقت) استفاده می گردد.

➤ ۳- انواع صحیح را می توان در انواع اعشاری ریخت ولی برعکس آن درست نمی باشد.

```
Var
i: integer;
c: Real;
```

```
begin
i:=123; → درست
c:=236; → درست
i:=12.36; → خطای کامپایلری
End.
```

۴-الف) داده‌های Char را نمی‌توان در هیچ یک از انواع صحیح یا اعشاری یا بولین ریخت و برعکس (داده کارکتری را می‌توان در String ریخت ولی داده‌های String را نمی‌توان در Char ریخت).

ب) داده‌ی String را نمی‌توان در هیچ کدام از دیتا تایپ‌ها ریخت و برعکس.

ج) داده‌های boolean را نمی‌توان در هیچ یک از دیتا تایپ‌ها ریخت و برعکس.

✚ اگر در ابتدای برنامه رهنمود $\{R+\}$ را بنویسیم آن‌گاه هنگام اجرا اگر خطای سرریزی رخ دهد (خطای منطقی) پیام خطایی صادر می‌دهد (در هنگام اجرا نه در هنگام کامپایل).

* اپراتورهای ریاضی

+ (به اضافه)، - (تفریق)، * (ضرب)، / (علامت تقسیم: جواب آن همیشه عدد اعشاری می‌باشد حتی اگر دو عملوند صحیح باشند)، div جواب تقسیم همیشه صحیح می‌باشد، mod باقی مانده‌ی تقسیم را بر می‌گرداند.

✚ Mod و div فقط برای عملوندهای صحیح کاربرد ندارند و اگر یکی یا هر دو عملوند اعشاری باشد خطای کامپایلری داده می‌شود.

✚ اگر یک یا دو عملوند div و mod عددی منفی باشند.

۱) علامت div از ضرب علامت دو عملوند به دست می‌آید. $-12 \text{div} -3 \rightarrow +4$

۲) علامت mod همواره هم علامت مقسوم یا عملوند سمت چپ mod می‌باشد:

$-12 \text{mod} -3 \rightarrow -4$

$12 \text{mod} -3 \rightarrow +4$

✚ در عملگرهای +، -، * اگر حداقل یکی از عملوند اعشاری باشد حاصل اعشاری است.

✚ عملگرهای ذکر شده در بالا عملگرهای ریاضی دوتایی (دو عملوندی) بودند.

✚ علامت + یا علامت - عملگرهای یکتایی ریاضی می‌باشند می‌توان به هر تعداد عملگرهای + یا - پشت سر هم قرار داد. در این حال مشابه ریاضیات این علامت‌ها را در هم ضرب می‌کنیم: مثال:

$b:=+a-3;$

خروجی $\leftarrow -1$ $c:=4+5;$

✚ هنگام تایپ عبارات طرفین DIV و MOD باید حداقل یک فاصله خالی قرار داد وگرنه خطای کامپایلری رخ می‌دهد.

Var

L,H,C,i:integer;

d,b:Real;

begin

i:=12'

b:=3.5;

c:=4;

d:=i/c; $\rightarrow 12/4 \rightarrow 3.5$

خطا چون متغیر اعشاری نمی‌تواند وارد متغیر صحیح شود $\rightarrow 12/4 \rightarrow 3.5$ $L:=i/c;$

$L:=\text{imod } c; \rightarrow 12/4 \rightarrow 0$

$d:=b+c; \rightarrow 3.5+4 \rightarrow 7.5$

$L:=c*i; \rightarrow 48$

$H:=I \text{ div } b; \rightarrow$ خطای کامپایلری صحیح باشند div باید هر دو عملوند طرفین

$H:=C--++i; \rightarrow +16$

End.

فصل

۴

دستورات ورودی و خروجی

۳۷

* دستورات write و writeln: (پروجرهای از پیش تعریف شده هستند)

- از این دستورات برای چاپ محتوی متغیرها، نتایج عبارات، مقدار ثابت‌ها یا خروجی توابع یا هر پیام دلخواه که پیام‌ها در داخل علامت تک گیومه نوشته می‌شود و برای چاپ علامت گیومه باید دو تا علامت تک گیومه جلوی دستورات چاپ بنویسیم:
- Write ('Ali'); → Ali
 Writeln ("Ali"); → 'Ali'
- در دستور Write مکان‌نما پس از آخرین کارکتر که چاپ می‌کند در همان خط می‌ماند ولی در دستور Writeln مکان‌نما بلافاصله به خط بعد می‌رود یعنی یک خط خالی رد می‌کند و در این دستورات می‌توان همزمان چندین عبارت را که با کاما از هم‌دیگر جدا شده‌اند را چاپ کرد.
- جمله آخر فرقی نمی‌کند Write باشد یا Writeln چرا که Write و Writeln بودن وضعیت چاپ بعدی را مشخص می‌کند نه چاپ فعلی.
- محتوی متغیرهای منطقی و عبارات منطقی را می‌توان با دستورات خروجی چاپ کرد در این صورت اگر درست باشد TRUE و اگر غلط باشد FALSE چاپ می‌شود (با حروف بزرگ چاپ می‌شوند).
- با دستورات Write و Writeln نمی‌توان محتویات متغیرهای شمارشی، مجموعه‌ای و اشاره‌گر را چاپ کرد.
- ۱- کد #7 ← بیپ کامپیوتر را به صدا در می‌آورد (alrb): → Write (#7);
- ۲- کد #10 ← مکان‌نما را در همان ستون به خط بعد می‌برد:

Hello :

Write ('Hello:',#10,'Ali') →

Ali

۳- کد #13 ← مکان‌نما را به اول همان خط می‌برد (Carriage rebvrm): Write ('Hello',#13,'Ali') →

Hello:#13Ali → AliLo →

می‌توان در آرگون اول دستورات خروجی نوع دستگاه خروجی را تعیین کرد مثلاً اگر out put گذاشته شود نشان‌گر دستگاه خروجی استاندارد که اغلب مونیتر می‌باشد و اگر Lst گذاشته شود نمایان‌گر چاپ‌گر می‌باشد اگر نام دستگاه خروجی نوشته نشود همان مونیتر فرض خواهد شد برای استفاده از چاپ‌گر باید یونیت printer را uses کرد:

User printer;

Begine

Write (Lst,'shallo'); → (روی کاغذ) Shallo در چاپ‌گر چاپ می‌شود

End.

فصل

۵

دستورات کنترلی

* If then S1 else S2; شرط

۴۶

۱- اگر شرط جلو if درست باشد جمله S1 و اگر غلط باشد جمله جلو else یعنی S2 چاپ می‌شود (اجرا می‌شود).

۲- شرط جلوی if می‌تواند داخل پرانتز باشد یا نباشد.

۳- اگر در جای شرط عملگرهای منطقی (Not, XOR, AND, OR) را به کار ببریم باید طرفین آن در داخل پرانتز نوشته شود.

۴- جمله قبل از else، نقطه ویرگول نمی‌خواهد. در صورت داشتن خطا کامپایلری است.

در صورت درست بودن شرط S1 اجرا می‌شود → If then S1;

کلمه ok در هر صورت اجرا خواهد شد → Write ('ok');

۵- اگر دستورات جلوی then یا else بیش‌تر از یک جمله بوده از begin و End استفاده می‌کنیم.

* ifهای تو در تو (Nested ifs)

در ifهای تو در تو هر else یه نزدیک‌ترین IF ای که در همان بلوک باشد و قبل از آن if به else دیگری مربوط نشده باشد ارتباط

داده می‌شود.

در برنامه روبرو اگر $x=3$ و $y=6$ پیام LALO روی صفحه نمایش چاپ می‌شود در غیر این صورت پیام XALLO

چاپ می‌شود. اگر $X=0$ باشد بدون توجه به مقدار y خروجی نخواهیم داشت چون else مربوط به $y=2x$ است و اگر

$x=0$ بدون توجه به y پیام جلوی else دومی یعنی poore چاپ می‌شود.

if x= 3 then

if y= 2x then Write ('LALo')

else write ('Xallo');

End.

else write ('poore')

End.

فصل

۶

آرایه و رشته

۵۷

۱- متغیر اندیس‌دار:

آرایه‌ها متغیر هستند پس در قسمت var تعریف می‌شود و می‌توان در قسمت Type یا Const اسمی مجازی برای آن انتخاب یا آن را مقداردهی کرد.

۲- مجموعه‌ای از متغیرهای یک نوع (برخلاف رکورد (Record))

۳- هنگام ذخیره تمام خانه‌های آن پشت سر هم در حافظه قرار می‌گیرند طوری که در آدرس کم‌تر اولین عنصر در آدرس بیش‌تر آخرین عنصر آرایه ذخیره می‌شود و هر عنصر توسط اندیس آن در آرایه تعیین می‌شود.

۴- آرایه می‌تواند یک بعدی یا چند بعدی باشد (ماتریس).

۵- قاعده نام‌گذاری: مشابه متغیرهای معمولی.

Var

a: Array [1...6] of Integer;

a → نام متغیر

عدد ۱۲ را در خانه شماره دوم می‌ریزد → a[2] := 12

تعداد خانه‌ها = 6 - 1 + 1 = 6

a[1] ...	... a[6]				
11	12	23	34	45	56

* آرایه یک بعدی

طرز تعریف:

; نوع آرایه of [اندیس بالا ... اندیس پایین] Array : نام متغیر آرایه

تعداد خانه‌های آرایه از فرمول زیر به دست می‌آید:

۱+ اندیس پایین - اندیس بالا

➤ اندیس آرایه در پاسکال می‌تواند هر نوع اردینال (کارکتری - بولین - صحیح) باشد ولی نمی‌تواند اعشاری یا رشته‌ای باشد ولی در خانه‌های آرایه می‌توان اعداد اعشاری ریخت.

➤ هنگام تعریف آرایه اندیس پایین باید کوچک‌تر از اندیس بالا باشد وگرنه خطای کامپایلری صادر می‌شود. ولی اگر اندیس بالا و پایین برابر باشد آن وقت آرایه‌ی تک خانه‌ای داریم.

فصل

۷

زیر برنامه‌ها (تابع و پروسجر)

ماژولار یا پیمانه‌ای بودن زبان‌های ساخت‌یافته ← یعنی هر برنامه را می‌توان به برنامه‌های فرعی کوچک‌تر (زیر برنامه) تجزیه کرد.

اهمیت برنامه‌های فرعی کوچک‌تر:

- ۱- کمک می‌کند تا برنامه اصلی طولانی نباشد.
- ۲- نوشتن و آزمایش کردن یک قطعه برنامه ساده‌تر از یک برنامه بزرگ است.
- ۳- باعث جلوگیری از تکرار دستورات مشابه در برنامه می‌گردد.
- ۳- باعث جلوگیری از تکرار دستورات مشابه در برنامه می‌گردد. یک زیر برنامه یک‌بار در برنامه نوشته شده و می‌توان n بار آن را صدا زد.
- ۴- موضوع برنامه بیش‌تر می‌شود.

* انواع زیر برنامه در پاسکال

۱- تابع (Function)

۲- پروسجر (Procedure)

* تابع (Function)

- توابع را باید بالا تست اصلی (بیرون begin و End برنامه اصلی) تعریف کرد.
- متغیرهای درون تابع متغیرهای محلی هستند و فقط در درون تابع اعتبار دارند ولی متغیرهای بیرون تابع متغیرهای سراسری هستند و در کل برنامه معتبرند.
- تعریف تابع:

نوع خروجی تابع: (نوع پارامتر: ورودی‌های تابع) نام تابع Function

مثال $\rightarrow$ Function 2 (a,b: integer): integer;

- تابع فقط یک خروجی دارد و هم نام تابع می‌باشد.
- تابع می‌تواند ورودی نداشته باشد و یا یک یا چند ورودی داشته باشد.
- نام متغیرهای محلی با نام متغیرهای سراسری و نام پارامترها با نام آرگومان‌ها می‌تواند یکسان باشد.

فصل

۸

گونه شمارشی، مجموعه و رکورد

* متغیرهای شمارشی (Enumeration)

- ۱- از نوع اردینال هستند پس با توابع Suc, pred و Ord به کار برده می شوند و می توان جلوی Case آورده شود و می توان از آن ها در اندیس حلقه و آرایه استفاده کرد.
- ۲- متغیر هستند پس می توان از آن ها در Type اسم مجازی انتخاب کرد و بعد آن را در قسمت Var به کار برد.
- ۳- آن ها را می توان به صورت زیر قلمرو به کار برد و قواعد زیر قلمروها در مورد این متغیر صدق می کند.

تعریف

مجموعه ای از اعداد صحیح نام دار (مجموعه ای است مرتب که برنامه نویس آن را در قسمت Var یا Type نام گذاری می کند و از طرف چپ به راست آن ها را شماره دهی کرده که شماره دهی از صفر شروع شده و باید عناصر داخل پرانتز باشند و هم چنین در نام گذاری آن ها نمی توان عدد، '، ' و را به کار برد و در دستورات نمی توان مقدار بعد از آخرین یا مقدار قبل از اولین نام گذاری را جایگزین کرد و مقدار دهی در این صورت خطای کامپایلری یعنی نمی توان نوع خارج از پرانتز یا زیر قلمرو را به کار برد و اگر دو گونه شمارش مجزا دارای عناصر (مقادیر) یکسان باشند خطای کامپایلری می باشد و فضایی که این متغیر اشغال می کند ۱ بایت می باشد و گونه های شمارش حداکثر ۲۵۶ عنصر دارند).

متغیرهای شمارش را نمی توان با Read خواند یا با write چاپ کرد و همچنین نمی توان آن ها را در عبارات ریاضی آورد. و باید برای خواندن و نوشتن آن ها از روش غیرمستقیم استفاده کنیم. به صورتی که متغیرهای شمارش هم نوع را به یکدیگر نسبت می دهیم و یا آن ها را با هم مقایسه می کنیم.

(داده های شمارشی را می توان در عبارات شرطی آورد).

از گونه های شمارشی می توان به عنوان پارامترهای تابع و پروسجر (چه خروجی و چه ورودی) استفاده کرد ولی باید در قسمت Type به فرم کلی آن اسم مجازی داد و در غیر این صورت خطای کامپایلری می باشد (مثل آرایه ها).

Var

JIN: (Ako , LALO, MAMO, Poore); → درست

JN:(Diar, shaho, "Mar", '2A' , 3B); → (چون عدد '، ' نمی توان وارد شود) خطا

a,b:integre

فایل‌ها در پاسکال

فصل

۹

انواع داده‌های `boolean`, `Char`, `Real`, `Integer`، رشته، آرایه، رکورد، مجموعه، شمارشی و ... در حافظه RAM ساخته می‌شوند و با خروج از برنامه یا خاموش شده دستگاه اطلاعات آن‌ها از بین می‌رفت ولی ساختمان داده دیگری به نام فایل (`file`) وجود دارد که هنگام خروج از برنامه و یا خاموش شدن دستگاه اطلاعات آن از بین نمی‌رود و در حافظه جانبی مثل دیسک ساخته می‌شود.

* انواع فایل

۱- متنی (`Text`)

۲- نوع داده (`Typed`)

۳- بدون نوع (`untyped`)

فایل‌ها متغیر هستند پس می‌توان برای آن در قسمت `Type` اسمی مجازی تعیین کرد و آن را در قسمت `Const` مقداردهی کرد.

* فایل‌های متنی (`Text file`)

۱- عموماً برای ذخیره متن‌های `Asc II` مثل نامه‌ها و توضیحات که پردازش‌هایی مثل مرتب‌سازی یا حذف در آن‌ها غالباً انجام نمی‌گیرد.

۲- اندازه خطوط در این فایل یکسان نیست و انتهای خطوط با دو کد `#13` و `#10` مشخص می‌کند. (`#13`: مکان‌نما را به اول همان

خط می‌برد: `CR=carriage Return`, `#10`: مکان‌نما را در همان ستون به خط بعدی می‌برد: `LF=Line Feed`)

۳- نحوه‌ی دستیابی به این نوع فایل ترتیبی یا سریالی است یعنی برای دستیابی به خط ۱۰ باید ۹ خط اول را خواند.

۴- در فایل‌های متنی برای ذخیره اعداد به جای هر رقم کداسکی آن ذخیره می‌شود پس برای هر رقم یک بایت فضا گرفته می‌شود.

چهار بایت `$34`, `$36`, `$37`, `$32` → 2764 →

۵- می‌توان محتویات فایل متنی را بر روی هر ویرایش‌گر استاندارد مشاهده کرده تصویر آینه‌وار وارد حافظه نمی‌باشد.

۶- هنگام ذخیره داده‌ها بر روی فایل متنی ممکن است تبدیلاتی انجام گیرد و به همین علت تبدیلات و سریالی بودن فایل متنی

نسبت به فایل نوع‌دار کندتر است.

فصل

۱۰

اشاره گرها

۱۱۱

اشاره گر (Pointers): متغیری که در آن آدرس یک خانه می‌نشیند. اشاره گر متغیر است پس می‌توان در قسمت Const مقداردهی شود و یا در قسمت Type اسمی مجازی برای آن به کار برد:

Const Pi: ^ Integer = @i

نحوه تعریف اشاره گر مانند سایر متغیرهاست ولی با این تفاوت که بعد از دو نقطه یا قبل نوع دیتا تایپ متغیر اشاره گر علامت می‌آید:

Var

P اشاره‌گری است که در آن آدرس یک متغیر اعشاری قرار می‌گیرد و هنوز به جایی اشاره نمی‌کند. $P: ^ \text{Integer} \rightarrow$ برای مقداردهی اشاره گر چه در داخل برنامه و چه در قسمت Const باید بعد از علامت =: در برنامه اصل و علامت = در Const یا قبل از متغیری که آدرس آن در داخل متغیر اشاره گر قرار می‌گیرد علامت @ بیاید و اگر نه اشاره گر به هیچ جا در برنامه اشاره نمی‌کند و خطای منطقی رخ می‌دهد و یا سیستم قطع می‌شود.

Const

P: ^ Integer = @i ;

Var

.

.

.

Begin

P := @ x ; $\rightarrow$ آدرس متغیر x در اشاره گر P ریخته می‌شود

K := P

End.

هنگام مقداردهی اشاره گرها باید متغیری که در متغیر اشاره گر ریخته می‌شود نوع آن‌ها با هم یکسان باشد در غیر این صورت خطای منطقی رخ می‌دهد.

اشاره گرها در زبان پاسکال از نوعی که باشد ۴ بایت فضا اشغال می‌کنند. چون اشاره گرهای پاسکال هر دو قسمت آدرس سگمنت و آفست سیستم عامل Dos را ذخیره می‌کنند.

بخش دوم

زبان C

فصل

۱

ساختار برنامه C

۱۲۱

- ✦ زبان C مانند زبان پاسکال همه منظوره می باشد.
- ✦ هر بلوک در زبان C مابین دو علامت {,} قرار می گیرد و در حکم یک جمله می باشد.
- ✦ منطق برنامه C و پاسکال از بالا به پایین است و می توان هسته ی اصلی برنامه رادر یک تابع نوشت و وظایف فرعی را به صورت صدا زدن توابعی در قسمت اصلی انجام داد.
- ✦ زبان C و پاسکال هر دو دارای قابلیت Top down programming و modular programming هستند.
- ✦ Top down programming یعنی تا شی اعلان نشود نمی توان از آن ها استفاده کرد و برنامه را می توان طوری نوشت که از پرش به بالا استفاده نشود.
- ✦ modular programming: یعنی برنامه را به قسمت های مجزا تقسیم کرده و اطلاعات هر قسمت از دید قسمت دیگر مخفی است و این خاصیت باعث می شود نوشتن برنامه ساده تر و وضوح آن بیش تر شود و عیب یابی (debug) برنامه را ساده تر می کند.
- ✦ زبان C قابل حمل (portable) است. یعنی source برنامه را می توان بر روی انواع کامپیوترها و سیستم عامل ها کامپایل کرده و کد ماشین دستگاه خاص را ایجاد کرد.
- ✦ زبان C یک زبان سطح میانی (متوسط) یا middle-level language می باشد. یعنی هم دارای قابلیت زبان های سطح بالا مثل پاسکال و هم دارای قابلیت زبان های سطح پایین مثل اسمبلی می باشد.
- ✦ زبان C و پاسکال به صورت کامپایلر می باشد که در این زبان ها کل برنامه خوانده شده تبدیل به کد ماشین شده و بعدا اجرا می گردد. (ترجمه و اجرا مجزا از همدیگر)
- ✦ زبان بیسیک مفسری خوانده می شود که در زبان های مفسری ابتدا خط خوانده شده و اگر بدون خطای گرامری باشد تبدیل به زبان ماشین می شود.
- ✦ مفسری: ترجمه و اجرا همزمان و خط به خط صورت می گیرد.
- ✦ در زبان C حروف کوچک و بزرگ با هم فرق دارند. print معادل PRINT نیست. (برعکس پاسکال)
- ✦ متغیر در ابتدای برنامه نوشته و تعریف می شوند و تعریف آن ها به صورت زیر است:

int s: نوع صحیح

float z: نوع اعشاری

char c: نوع کارکتری

فصل

۲

متغیرها و ثابت‌ها

- ✚ نام شناسه در توربو C می‌تواند ۱ تا ۳۲ کارکتر باشد و در C++ نام یک متغیر یا شناسه می‌تواند ۱ تا ۱۰۲۴ کارکتر باشد.
- ✚ تعداد کلمات کلیدی در زبان C نسبتاً کم است.
- ✚ بعضی از شناسه‌های استاندارد در C ← mian, print f, scan f, sin, cos, exp, sqrt و

انواع داده‌ها در زبان C (datatype)

- ۱- int a; (دیتا تایپ صحیح): ۲ بایت فضا اشغال می‌کند. ۳۲۷۶۷+ تا ۳۲۷۶۸-
- ۲- Float r,s; (دیتاتایپ اعشاری): اعشاری در c ۴ بایت فضا اشغال می‌کند و در پاسکال ۶ بایت.
- ۳- char l; (دیتاتایپ کارکتری): ۱ بایت حافظه اشغال می‌کند. ۱۲۷+ تا ۱۲۸-
- ۴- double s; (دیتاتایپ اعشاری با دقت زیاد): ۸ بایت حافظه اشغال می‌کند.
- ۵- void s; دیتاتایپی که هیچ فضایی اشغال نمی‌کند و هیچ چیزی را ذخیره نمی‌کند. مورد استفاده:
- ۱- برای تعریف اشاره‌گر عمومی (pointer)
- ۲- برای معرفی تابعی که چیزی برنمی‌گرداند (خروجی ندارد)
- ۳- برای معرفی تابعی که چیزی نمی‌گیرد (پارامتر (ورودی) ندارد).
- ✚ اغلب برنامه‌نویسان اگر دامنه مقادیر اجازه بدهد به جای متغیر صحیح int دوتایی از متغیر کارکتری char یک بایتی استفاده می‌کنند. چون دسترسی به یک بایت سریع‌تر از دسترسی به ۲ بایت است.

پیشوندهای تغییر دهنده نوع (type modifier)

- ۱- signed: فقط با int و cahr می‌آید و اگر آن را پشت float یا double بنویسیم خطای کامپایلری صادر می‌شود. علامت‌دار ← یعنی می‌توان عدد مثبت یا منفی را در خود ذخیره کند.
- signed int a; (درست)
- signed float; (خطا)
- ۲- unsigned ← فقط با int و cahr می‌آید و اگر آن را پشت float و double بنویسیم خطای کامپایلر می‌باشد. فقط علامت مثبت را ذخیره می‌کند و محدوده آن افزایش پیدا می‌کند.
- ۳- short ← می‌تواند پشت هر ۴ نوع دیتاتایپ (int, float, double, cahr) قرار گیرد.

بانک سوالات

جمع آوری سوالات کنکور کاردانی به کارشناسی
کارشناسی ارشد و دکتری به صورت فصل بندی شده

سوالات تألیفی

تألیف سوالات مشابه کنکور

جامع

حاوی تمامی مطالب و نکات لازم
برای کنکور براساس منابع

الگوریتم

چکیده‌ی تمامی مطالب و نکات لازم
برای کنکور براساس منابع



دریافت نمونه‌ی کتاب به صورت رایگان



www.agkins.com

جامع

حاوی تمامی مطالب و نکات لازم
برای کنکور براساس منابع

بانک سوالات

جمع آوری سوالات کنکور کاردانی به کارشناسی
کارشناسی ارشد و دکتری به صورت فصل بندی شده

الگوریتم

چکیده‌ی تمامی مطالب و نکات لازم
برای کنکور براساس منابع

سوالات تألیفی

تألیف سوالات مشابه کنکور



دریافت نمونه‌ی کتاب به صورت رایگان



www.agkins.com

شماره تماس با نمایندگی‌های فعال و رسمی آناهید گستر خلیلی

شماره تماس	نمایندگی	شماره تماس	نمایندگی
۰۹۱۰۱۷۱۱۸۷۲	سبزوار (آقای شریفان)	۰۹۹۱۳۷۱۳۷۵۰	اراک (آقای حسینی)
۰۹۹۱۳۷۱۳۷۴۱	شاهرود (آقای سلمانی)	۰۹۱۹۹۱۰۱۲۴۸	اردبیل (خانم عاصمی‌زاده)
۰۹۱۹۹۱۰۱۲۴۹	شهرکرد (خانم تقی‌پور)	۰۹۱۹۹۱۰۱۲۴۱	ارومیه (آقای آرمیون)
۰۹۱۹۵۷۳۰۱۵۲	شیراز (آقای فروردین / خانم هوشمندی)	۰۹۱۹۶۳۲۱۸۵۲	اسفراین (خانم اسماعیل‌زاده)
۰۹۱۹۳۷۱۳۷۴۲	فسا (خانم شعبانی)	۰۹۱۹۵۷۳۰۱۵۰	اصفهان (آقای کیانی)
۰۹۱۹۵۷۳۰۱۴۹	قزوین (خانم چگینی)	۰۹۱۹۵۷۳۰۱۵۴	اهواز (آقای دکتر رضازاده)
۰۹۱۹۷۷۸۱۹۴۷۱	کرج (آقای دکتر علیرضاپور)	۰۹۱۹۰۹۸۰۳۱۸	ایلام (آقای دکتر بوچانی)
۰۹۱۳۱۹۸۱۲۷۰	کرمان (خانم ضمانتی‌یار)	۰۹۱۹۶۸۵۳۱۱۶	بروجرد (آقای احمدی‌صدر)
۰۹۱۹۵۷۳۰۱۴۸	کرمانشاه (آقای ابراهیمی)	۰۹۹۱۳۷۱۳۷۴۹	بم (خانم سرحدی‌نژاد)
۰۹۱۹۵۳۷۱۸۹۰	کوهدشت (خانم یاریان)	۰۹۱۹۵۹۰۷۲۰۳	بیرجند (آقای دکتر بهروان)
۰۹۱۹۹۱۰۱۲۴۷	گرگان (آقای شاه‌علی)	۰۹۱۹۵۷۳۰۱۴۷	تبریز (خانم عاصمی‌زاده)
۰۹۱۹۵۷۳۳۱۷۹	مراغه (آقای صمدی)	۰۹۰۱۳۷۳۷۸۹۶	جهرم (آقای یاعلی جهرمی)
۰۹۹۱۳۷۱۳۷۴۴	مشهد (خانم دکتر شجاعی)	۰۹۱۹۹۱۰۱۲۴۰	جیرفت (خانم دکتر محمدی)
۰۹۱۹۵۷۳۳۱۷۹	میاندوآب (آقای صمدی)	۰۹۱۹۲۷۰۵۸۷۸	خرم‌آباد (خانم میر)
۰۹۱۹۶۳۵۰۷۶۸	نیشابور (آقای موسوی)	۰۹۱۹۵۷۳۰۱۵۳	رشت (خانم تنها)
۰۹۱۹۵۷۳۳۲۸۴	هرمزگان (آقای زارعی)	۰۹۱۹۶۸۲۹۲۸۰	رفسنجان (آقای یوسفی)
۰۹۱۹۵۷۳۰۱۵۵	همدان (آقای سوری)	۰۹۱۹۹۱۰۱۲۴۵	زاهدان (آقای مهمان‌دوست)
۰۹۱۹۵۹۰۷۲۰۴	یاسوج (آقای مرتضوی)	۰۹۱۹۲۷۰۵۸۷۱	زنجان (خانم دکتر هوشیار)
۰۹۱۹۹۱۰۱۲۴۳	یزد (خانم آزاد)	۰۹۱۹۷۷۸۱۹۴۴	ساری (آقای دکتر اکبری)

بانک کتاب ناهید



«هر کتابی، از هر انتشاراتی را از ما بخواهید»

✓ جامع‌ترین بانک کتاب

✓ تحویل روزانه

✓ ارسال به تمامی نقاط کشور

✓ سفارش کتاب به صورت تلفنی و آنلاین

www.NIBS.ir



کتاب دانشگاهی، فنی و مهندسی، علوم پزشکی، علوم انسانی، عمومی،
ادبی، مذهبی، کمک آموزشی، کودک و نوجوان و کتاب نفیس

فروشگاه: تهران - خیابان انقلاب - روبه روی درب اصلی دانشگاه تهران

پاساژ فروزنده - طبقه همکف - پلاک ۳۳۱

تلفن: ۶۶۴۸۹۳۷۵ - ۰۲۱ - ۶۶۴۸۹۳۴۹ - ۰۲۱