

اصول برنامه نویسی

درسنامه‌ی تستی تشریحی

گردآورنده 

محمد حسینی روندی

سرشناسه	حسینی روندی، محمد، ۱۳۶۸ -
عنوان و نام پدیدآور	درسنامه اصول برنامه‌نویسی / نویسنده محمد حسینی روندی.
مشخصات نشر	تهران: انتشارات علمی سنا، ۱۳۹۵.
مشخصات ظاهری	۲۴۰ ص.: مصور، جدول، نمودار .
شابک	۹۷۸-۶۰۰-۸۴۵۶-۱۳-۱
وضعیت فهرست نویسی	فیفا
موضوع	برنامه‌نویسی -- راهنمای آموزشی (عالی)
موضوع	Computer programming -- Study and teaching (Higher)
موضوع	ساختار داده‌ها -- راهنمای آموزشی (عالی)
موضوع	(Data structures (Computer science) -- Study and teaching (Higher)
موضوع	انفورماتیک پزشکی -- راهنمای آموزشی (عالی)
موضوع	--Study and teaching(Higher)Medical informatics
موضوع	سی ++ (زبان برنامه‌نویسی کامپیوتر) -- راهنمای آموزشی (عالی)
موضوع	(C++ (Computer program language) -- Study and teaching (Higher)
رده بندی کنگره	Q۷۶/۶/ج۵د۴ ۱۳۹۵
رده بندی دیویی	۰۰۱/۶۴۲
شماره کتابشناسی ملی	۴۴۰۸۶۳۵



موسسه علمی انتشاراتی سنا

نام کتاب اصول و برنامه نویسی

گردآورندگان محمد حسینی روندی

شابک ۹۷۸-۶۰۰-۸۴۵۶-۱۳-۱

سال چاپ دوم - ۱۴۰۲

صفحه‌آرایی انتشارات علمی سنا

طراح جلد لیلا انصاری

پست الکترونیک elmisana@gmail.com

فروش اینترنتی sanabook.com

تیراژ ۱۰۰ نسخه

برای مشاهده قیمت اسکن کنید. 



دفتر مرکزی: تهران، میدان انقلاب، خیابان جمالزاده شمالی، خیابان فرصت شیرازی، پلاک ۷۲، طبقه همکف

تلفن: ۰۲۱۶۶۵۷۴۳۴۵-۶

مقدمه:

کتاب حاضر، تلاشی است برای کمک به دانش‌پژوهانی که قصد شرکت در آزمون ارشد یا دکتری انفورماتیک پزشکی را دارند. این کتاب به ویژه برای کسانی مناسب است که بچگونه دانش قبلی در مورد برنامه نویسی نداشته و از رشته‌هایی غیر از علوم کامپیوتر و حوزه‌های مربوطه آمده‌اند. با توجه به هدف اولیه کتاب، تأکید بر این موضوعات طولانی و خسته‌کننده پرسنل و تلاش شده است به جای حجم و قشور کردن کتاب به هدف اصلی آن پرداخته شود تا زمان و انرژی کمتری از خواننده برای یادگیری مطالب گرفته شود. لذا در راستای این مقصود، از زبان ابتدای کتاب به مطالبی پرداخته شده است که علاوه بر افزایش فهم خواننده از مطالب و محاسبات کلیدی زبان برنامه نویسی C، او را برای نتیجه‌گیری بهتر از آزمون آماده می‌سازد.

زبان برنامه نویسی C، یکی از زبان‌های پرکاربرد، وسیع و قدرتمند در زمینه ایجاد انواع برنامه‌های سیستمی و کاربردی است. لذا ذکر تمام مطالب و کاربردهای آن با توجه به هدف این کتاب، باعث حشو و سردرگمی خواننده شده و منجر به عدم کسب نتیجه‌بخش می‌گردد. لذا در این کتاب علاوه بر حفظ شیوایی کلام و سادگی سخن، برخلاف کتاب‌های درسی سعی شده است از مباحث غیر ضروری تا جایی که لطمه‌ای به یادگیری مفاهیم اصلی وارد نکند، پرهیز شود. همچنین سعی نگارنده بر این بوده است که مطالب به نحوی دسته‌بندی و فصل‌بندی گردد که فهم آنها برای خواننده راحت‌تر و در عین حال به محاسبات کلیدی، به ویژه برای موفقیت در آزمون‌های ارشد و دکتری انفورماتیک، تکمیل گردد. در انتهای هر فصل نیز، چکیده‌ای از مطالب و سوالات طبقه‌بندی شده، ارائه شده است.

با توجه به تجربیات بدست آمده از آزمون‌های ارشد و دکتری انفورماتیک پزشکی، هدف غایی نگارنده، فراهم آوردن منبعی مناسب هم برای یادگیری سریع و هم مرور جامع مطالب برای این آزمون بوده است. امید است که این کتاب مقبول خوانندگان گرامی واقع گردد. کلام آخر اینکه، با وجود تمام تلاش‌های صورت گرفته، مسلماً کتاب حاضر خالی از اشکال نیست. لذا از تمامی خوانندگان عزیز خواهشمندم، هرگونه انتقاد و پیشنهاد پیرامون مطالب این کتاب را به اینجانب منعکس نمایند.

محمد حسینی روفندی

فهرست:

فصل اول: مقدمات زبان C.....	۱
ویژگی‌های زبان C.....	۲
انواع داده‌ها.....	۷
پیشوندهای تغییر دهنده نوع (type modifier).....	۹
متغیرها.....	۱۱
ثابت‌ها.....	۱۵
عملگرها.....	۱۹
خلاصه فصل.....	۳۵
تست‌های طبقه‌بندی شده فصل اول.....	۳۶
پاسخ‌های تشریحی.....	۴۰
فصل دوم: ساختار برنامه، ورودی و خروجی.....	۴۶
ورودی و خروجی داده‌ها.....	۴۸
تابع printf().....	۴۸
تابع scanf().....	۶۲
خلاصه فصل.....	۷۲
تست‌های طبقه‌بندی شده فصل دوم.....	۷۳
پاسخ‌های تشریحی.....	۷۵
فصل سوم: ساختارهای کنترلی.....	۷۷
ساختارهای تصمیم.....	۷۸
دستور if - else.....	۷۸
ifهای تو در تو یا Nested ifs.....	۸۰
ifهای پلکانی.....	۸۲

۸۳	switch دستور
۸۷	حلقه‌های تکرار
۸۷	for حلقه
۹۰	while حلقه
۹۲	do-while حلقه
۹۳	حلقه‌های تو در تو
۹۳	continue و break دستورات
۹۴	goto دستور
۹۵	exit, return, abort دستورات
۹۷	خلاصه فصل
۹۸	تست‌های طبقه‌بندی شده فصل سوم
۱۰۸	پاسخ‌های تشریحی
۱۱۶	فصل چهارم: آرایه‌ها و رشته‌ها
۱۱۷	آرایه‌های یک بعدی
۱۲۰	آرایه‌های چند بعدی
۱۲۰	آرایه دوبعدی
۱۲۲	آرایه سه‌بعدی
۱۲۲	مقداردهی اولیه آرایه‌ها
۱۲۵	رشته‌ها
۱۲۹	توابع استاندارد کار با رشته‌ها
۱۳۲	خلاصه فصل
۱۳۳	تست‌های طبقه‌بندی شده فصل چهارم
۱۳۸	پاسخ‌های تشریحی

فصل پنجم: اشاره گرها..... ۱۴۲

متغیرهای اشاره گر..... ۱۴۳

اشاره گرها و انواع متغیرها..... ۱۴۵

عملیات بر روی اشاره گرها..... ۱۴۶

کاربرد اشاره گرها..... ۱۴۹

اشاره گر ثابت..... ۱۵۰

متغیرهای پویا..... ۱۵۰

رابطه اشاره گرها با آرایه ها و رشته ها..... ۱۵۲

خلاصه فصل..... ۱۵۵

تست های طبقه بندی شده فصل پنجم..... ۱۵۶

پاسخ های تشریحی..... ۱۶۱

فصل ششم: توابع و کلاس های حافظه..... ۱۶۶

توابع..... ۱۶۷

روش ارسال آرگومان ها به توابع..... ۱۷۱

ارسال آرایه ها به توابع..... ۱۷۲

اشاره گر به تابع..... ۱۷۳

توابع بازگشتی..... ۱۷۴

کلاس های حافظه و طول عمر متغیرها..... ۱۷۵

خلاصه فصل..... ۱۸۲

تست های طبقه بندی شده فصل ششم..... ۱۸۴

پاسخ های تشریحی..... ۱۸۷

فصل هفتم: ساختارها..... ۱۸۹

گونه های شمارشی (ترتیبی)..... ۱۹۰

استراکچر (ساختار).....	۱۹۳
ارسال استراکچر به تابع.....	۲۰۰
ارسال استراکچر به تابع با استفاده از اشاره گر (فراخوانی با مرجع).....	۲۰۱
لیست پیوندی (Linked List).....	۲۰۳
فیلد بیتی (Bit Field).....	۲۰۴
یونین (اتحاد) (Union).....	۲۰۷
خلاصه فصل.....	۲۱۰
تست های طبقه بندی شده فصل هفتم.....	۲۱۱
پاسخ های تشریحی.....	۲۱۳
پیوست ۱: تبدیل مبنایها.....	۲۱۵
تبدیل مبنای ۱۰ به مبنای ۲ و بلعکس.....	۲۱۷
تبدیل مبنای ۱۰ به مبنای ۸ و بلعکس.....	۲۱۹
تبدیل مبنای ۱۰ به مبنای ۱۶ و بلعکس.....	۲۲۰
پیوست ۲: کدهای اسکی.....	۲۲۲



فصل اول: مقدمات زبان C



زبان برنامه نویسی C یکی از معروفترین و پرکاربردترین زبانهای برنامه نویسی در جهان است. این زبان تکامل یافته‌ی زبان BCPL می باشد که از زبان B گرفته شده است. زبان C زبانی بسیار قدرتمند است و همین امر موجب شده که انواع برنامه های کاربردی، تجاری، آموزشی، اقتصادی، گرافیکی، بانک اطلاعاتی و حتی سیستم های عامل به این زبان نوشته شوند.

ویژگی های زبان C

زبانهای برنامه نویسی به سه رده کلی، زبانهای سطح پایین، زبانهای سطح میانی و زبانهای سطح بالا تقسیم بندی می شوند. زبانهای سطح پایین به آن دسته از زبانهای برنامه نویسی اطلاق می شود که قدرت بالایی در برنامه نویسی دارند، مانند دسترسی مستقیم به حافظه، توانایی کار با مفاهیم پایه مانند بیت، بایت، آدرس، امکان دسترسی و بهره برداری از سخت افزار و... . کار کردن با این زبانهای برنامه نویسی سخت و دشوار است چرا که از خوانایی پایینی برخوردارند و درک برنامه های نوشته شده به این زبانها برای انسان مشکل و زمانبر است. در مقابل زبانهای سطح پایین، زبانهای سطح بالایی همچون پاسکال قرار دارند. این زبانها از خوانایی بالایی توسط انسان برخوردارند. یعنی درک منطق برنامه ها و فهم کدهای نوشته شده در آن برای انسان آسان است چرا که کدها و ساختار برنامه های نوشته شده به این زبانها به زبان محاوره ای انسان نزدیک است. با این وجود این زبانها در مقابل زبانهای سطح پایین از قدرت چندانی برخوردار نیستند. زبان C از جمله زبانهایی است که تا حدود زیادی قدرت زبانهای سطح پایین و خوانایی کدهای زبانهای سطح بالا را دارد. لذا زبان C به عنوان یک زبان سطح میانی شناخته می شود. این زبان بسیار قدرتمند و انعطاف پذیر است و هر برنامه ای را می توان با آن نوشت.

جدول ۱-۱ انواع زبانهای برنامه نویسی (از نظر قدرت و خوانایی)

زبانهای سطح پایین	زبانهای سطح میانی	زبانهای سطح بالا
اسمبلر ماکرو اسمبلر	C C++ جاوا فورث	پاسکال ادا ماجولا-۲ کوبول بیسیک

از نظر ساخت یافتگی، زبانهای برنامه نویسی را به دو دسته کلی زبانهای ساخت یافته (ساختمند) و غیر ساخت یافته (غیر ساختمند) تقسیم می کنند. زبانهای ساخت یافته به آن دسته از زبانهای برنامه نویسی گفته می شود که با استفاده از مفهوم تابع برنامه به بخش های متعدد تقسیم می شود. در واقع هر تابع مسئول انجام یک فعالیت خاص در برنامه است. علاوه بر آن کدهای نوشته شده داخل بلاک هایی قرار می گیرند و برنامه بصورت بلاک به بلاک اجرا می گردد و



```
        goto error
    else
        :
    end if
end
:
error:
:
```

الف) احتمال آنکه این زبان ساخت یافته باشد، خیلی کم است.

ب) زبان فوق ساخت یافته (structural) است ولی طرز برنامه نویسی آن، به شکل ساخت یافته نبوده است.

ج) زبان فوق نمی تواند ساخت یافته باشد.

د) نمی توان در این مورد تصمیم گیری کرد.

بررسی گزینه ها:

گزینه الف: این گزینه صحیح نیست چرا که برنامه دارای بلاک بندی است (به علت استفاده از begin و end که یک بلاک از دستورات را تشکیل داده اند). بنابراین برنامه ساخت یافته است.

گزینه ب: همانطور که قبلا توضیح داده شد، در برنامه های ساخت یافته تا حد امکان از بکاربردن دستورات پرشی مانند goto اجتناب می شود چرا که احتمال خطا را افزایش داده و درک برنامه را سخت تر می کند. این طرز برنامه نویسی بیشتر مربوط به برنامه های غیر ساخت یافته است و با توجه به توضیحات گزینه الف این برنامه یک برنامه ساخت یافته است و بنابراین گزینه ب صحیح است.

گزینه ج: با توجه به توضیحات گزینه الف این گزینه غلط است.

گزینه د: این گزینه غلط است.

انواع داده ها

یک کامپیوتر به مسائل مانند یک انسان نگاه نکرده و مانند او فکر نمی کند. برای مثال دو عدد 45 و 45.0 با اینکه برابرند اما یکسان نیستند و برای کامپیوتر این دو عدد دو نوع متفاوت دارند، به عبارت دیگر جنس این اعداد باهم متفاوت است. کامپیوتر عدد اول را یک نوع عدد صحیح و عدد دوم را نوعی عدد اعشار می شناسد. به همین منظور در کامپیوتر برای انجام محاسبات مختلف و حل مسائل، انواع داده (data type) تعریف شده است که هر کدام وظیفه نگه داری یک نوع خاص از داده ها را بر عهده داشته و هر کدام محدوده ای دارند و میزان مشخصی از فضا را در حافظه اشغال می کنند. در زبان C پنج نوع داده اصلی وجود دارد که در جدول ۲-۱ آورده شده اند.



لذا به این نکات توجه داشته باشید. در زبان C اعدادی که بصورت روزمره با آنها سروکار داریم در مبنای ۱۰، اعدادی که با ۰ (صفر) شروع شوند، در مبنای ۸ و اعدادی که با 0x (صفر ایکس) شروع شوند، در مبنای ۱۶ در نظر گرفته می‌شود. برای مثال عدد 45 در مبنای ۱۰ بوده و لذا معادل با همان ۴۵ در نظر گرفته می‌شود، اما عدد 045 یک عدد در مبنای اُکتال بوده و معادل با ۳۷ می‌باشد و یا عدد 0x45 در زبان C به عنوان یک عدد در مبنای ۱۶ در نظر گرفته شده و برابر است با ۶۹ برای درک بهتر این موضوع بهتر "پیوست ۱" را در انتهای کتاب که پیرامون تبدیل مبناهای می‌باشد، مطالعه کنید.

مثال: تمام اعداد زیر در زبان C درست می‌باشد.

0x1 0565 342 0X20 02 0xab3

نکته

در مبنای هگزادسیمال از حروف (کوچک و بزرگ) a, b, c, d, e و f به جای اعداد ۱۰-۱۵ استفاده می‌شود.

نکته

توجه داشته باشید که در اعداد صحیح در مبنای هگزادسیمال، نقطه (.) و وجود ندارد و لذا استفاده از عددی مانند 0x2.5 باعث صدور پیام خطا خواهد شد. این عمل در اعداد صحیح مبنای دسیمال هستند، منجر به صدور پیام خطا نخواهد شد. اما قسمت اعشار آن (اعداد بعد از نقطه) حذف خواهند شد و این ممکن است منجر به ایجاد خطای منطقی گردد.

مثال: پس از اجرای دستور زیر، مقدار x عدد 76 خواهد بود و پیام خطایی صادر نمی‌گردد.

```
int x=76.34;
```

نکته

عددی مانند 02B منجر به صدور پیام خطا می‌شود، چرا که در یک عدد اُکتال b وجود ندارد و اگر مقصود، عددی در مبنای هگزادسیمال است می‌بایست در ابتدای آن 0x (صفر ایکس) یا 0X (صفر ایکس) گذاشته شود تا کامپایلر به آن عنوان یک عدد هگزادسیمال نگاه کند. در غیر اینصورت با پیام خطای روبرو خواهیم شد.



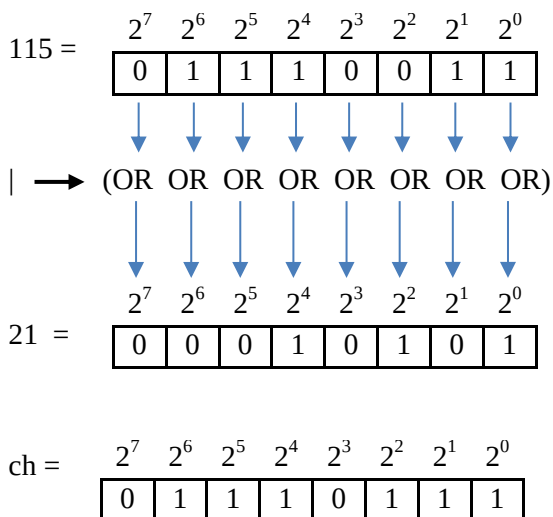
مثال:

```
unsigned char ch;
```

```
ch = 115 | 21;
```

```
printf("%d", ch);
```

در خط اول متغیری به نام `ch` از نوع کاراکتری بدون علامت تعریف شده است. در دستور دوم عدد ۱۱۵ با عدد ۲۱ با استفاده از عملگر `|` بیتی، OR شده و مقدار ۱۱۹ در متغیر `ch` ذخیره می‌شود.



دستور آخر یک تابع استاندارد در زبان C است که مقدار متغیر `ch` را بصورت عددی در صفحه نمایش کامپیوتر چاپ می‌کند.

عملگر $\wedge$ معادل XOR بیتی دو عملوند خود را برمی‌گرداند. مقدار این اپراتور هر جا که دو بیت مخالف هم باشند، یک (صحیح) و هر جا هر دو بیت با هم یکسان باشند صفر می‌باشد.

مثال:

$$115 \wedge 21 \rightarrow \begin{array}{r} 0111\ 0011 \\ \wedge \\ 0001\ 0101 \end{array}$$



“ نام فایل سرآیند با پسوند h. “ #include

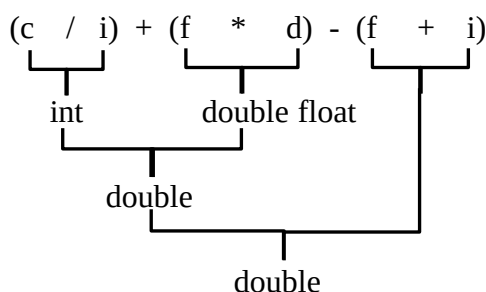
توجه داشته باشید که در انتهای دستورات پیش پردازنده (دستوراتی که با علامت # شروع می‌شوند) نباید علامت سمی‌کالن (;) قرار داد در غیر اینصورت خطای کامپایلری صادر خواهد شد.

۵. گزینه “ج” صحیح است.

قبل از انواع داده float و double نمی‌توان از modifierهای signed یا unsigned استفاده کرد. لذا گزینه “ج” نادرست است.

۶. گزینه “ب” صحیح است.

در عبارات محاسباتی در زبان C، هنگامی که دو متغیر تحت یک عملگر قرار می‌گیرند، حاصل آن در مناسب‌ترین نوع که در برگیرنده محدوده هر دو نوع متغیر باشد، قرار می‌گیرد. برای مثال اگر دو متغیر از انواع char و int با هم جمع شوند حاصل در محدوده int که محدوده‌ای مناسب برای ذخیره سازی عدد حاصل است، ذخیره می‌شود. لذا داریم:



بنابراین نتیجه نهایی در محدوده نوع double ذخیره می‌شود.

۷. گزینه “الف” صحیح است.

هنگامی که محتوای یک متغیر با محدوده بزرگتر در یک متغیر با محدوده کوچکتر ریخته می‌شود، بایت‌های با ارزشمندتر (بایت‌های سمت راست) که در محدوده متغیر مقصد نمی‌گنجند، از بین می‌روند. این امر خطای کامپایلری در پی ندارد ولی ممکن است خطای منطقی رخ دهد.



جدول ۳-۲ کاراکترهای فرمت و عملکرد آنها در دستور scanf()

کاراکتر فرمت	نوع اطلاعاتی که دریافت می شود
%c یا %C	تنها یک کاراکتر
%d یا %D	یک عدد صحیح دسیمال (علامتدار)
%i یا %I	یک عدد صحیح دسیمال، اکتال یا هگزادسیمال
%e یا %E	یک عدد اعشاری (بصورت ممیز شناور یا نماد علمی با حروف e یا E)
%f	یک عدد اعشاری (بصورت ممیز شناور یا نماد علمی با حروف e یا E)
%g یا %G	یک عدد اعشاری (بصورت ممیز شناور یا نماد علمی با حروف e یا E)
%o یا %O	یک عدد اکتال
%s	یک رشته از کاراکترها (عبارت رشته ای)
%u یا %U	یک عدد صحیح بدون علامت
%x یا %X	یک عدد مبنای هگزادسیمال با حروف کوچک یا بزرگ
%p یا %P	یک اشاره گر
%n	تعداد کاراکترهایی که تا قبل از این کاراکتر از ورودی دریافت شده اند، شمارش شده و در پارامتر متناظر با آن قرار می گیرد.
%[...]	خواند یک رشته از کاراکترهای معین (توضیح داده خواهد شد)
%%c یا %*C	صرف نظر از یک کاراکتر ورودی
%%	یک کاراکتر % از ورودی خوانده شده ولی جایی ذخیره نمی شود. به عبارت دیگر تابع printf() کاراکتر % را می خواند و از آن صرف نظر می کند.

نکته

توجه داشته باشید که در دستور scanf()، برخلاف دستور printf()، در برخی از کاراکترهای فرمت می توان از کاراکترهای بزرگ نیز استفاده کرد. برای مثال می توان از %D برای خواندن یک عدد صحیح استفاده کرد در حالیکه در دستور printf() فقط %d وجود دارد.

همانند دستور printf() در دستور scanf()، در عبارت اول کاراکترهای فرمت و در عبارت دوم به تعداد کاراکترهای فرمت، آدرس یک متغیر از نوع داده متناظر با آن قرار می گیرد.

مثال:

```
char ch; int i; float f;
```



```
2- int n;  
3- cin >> n;  
4- int p = 1;  
5- while (p <= n)  
6-     p*=2;  
7- while (n > 1) {  
8-     p/=2;  
9-     if(n>=p){  
10-         cout << 1;  
11-         n-=p;  
12-     }  
13-     else  
14-         cout << 0;  
15- }  
16- }
```

الف) خط (5) به `while(p > n)` تغییر کند.

ب) خط (7) به `while(p > 1)` تغییر کند.

ج) خط (8) یعنی `p /= 2` به قبل از خط (15) برده شود.

د) خط (11) به `n%=p` تغییر کند.

۶- اگر بخواهیم با استفاده از حلقه زیر بزرگترین مقسوم علیه مشترک `n` و `m` را بدست آوریم. کدام دو

سطر از این قطعه کد باید جابجا شوند؟

```
1- do {  
2-     m = r;  
3-     n = m % n;  
4-     r = n;  
5- } while (n!=0);
```

الف) ۲ و ۳ ب) سطر ۴ باید به قبل از شروع حلقه (قبل از `do`) منتقل شود. ج) ۲ و ۴ د) ۳ و ۴

۷- خروجی تکه برنامه زیر چیست؟

```
int x=24;  
if(x>0)  
if(x>20)  
if(x>40)  
printf("A");  
else  
printf("B");  
else  
printf("C");  
else  
printf("D");
```



پاسخ‌های تشریحی

۱. گزینه "ب" صحیح است.

با دستور switch می‌توان ifهای پلکانی را که در آن فقط شرط تساوی مورد بررسی قرار می‌گیرد، به سادگی و با یک کد کارآمدتر پیاده کرد. ifهای پلکانی که سایر شرایط را بررسی می‌کنند، نمی‌توان با دستور switch نوشت.

۲. گزینه "ج" صحیح است.

از آنجایی که در سوال صرفاً عدد نزدیک به خروجی برنامه خواسته شده است و از آنجایی که در هر بار اجرای حلقه، مقدار i که یک مقدار ثابت است، بر مضربی از ۲ که هر بار بزرگتر می‌شود، تقسیم می‌گردد. به عبارت دیگر در هر بار اجرای حلقه مقدار بسیار کمتری نسبت به قبل به مقدار sum اضافه می‌شود. بنابراین با تعداد معدودی اجرای حلقه مثلاً سه یا چهار بار، می‌توان تشخیص داد که مقدار نهایی sum به کدامیک از گزینه‌ها نزدیکتر خواهد بود.

۳. گزینه "ب" صحیح است.

در حلقه while شرط دستور $if((n \& 1) == 1)$ فقط زمانی برقرار خواهد بود که مقدار n یک باشد. لذا دستور $t=t*x$ فقط یکبار اجرا خواهد شد. بنابراین فقط دو دستور $t=t*x$ و $n/=2$ باقی می‌مانند. دستور $n/=2$ شرط حلقه را تغییر می‌دهد یا به عبارت دیگر شمارنده حلقه می‌باشد. دستور $x=x*x$ در هر بار مقدار x را در خودش البته با حفظ مقدارهای پیشین، ضرب می‌کند و در نهایت حاصل در t ریخته می‌شود.

۴. گزینه "الف" صحیح است.

در مواردی مانند این سوال، در صورتیکه درک منطق برنامه کمی مشکل و زمانبر است. کفایت که برنامه مورد نظر را با اعداد کوچک مانند ۲ و ۳ اجرا کنید. در این حالت سریعتر به جواب خواهید رسید.

۵. گزینه "ب" صحیح است.

می‌توان برنامه را با یک عدد که تنوع بیتی آن بالاتر است، اجرا کرد. برای مثال عدد ۲۰ که در مبنای دو به صورت ۱۰۱۱۰ است. اما از اعدادی که تنوع بیتی پایین دارند مانند ۲، ۴، ۸ و ... استفاده نکنید چرا که ممکن است شما را به اشتباه بیاورند.

۶. گزینه "ج" صحیح است.

این برنامه الگوریتم استاندارد است که با استفاده از روش اقلیدوسی بزرگترین مقسوم علیه مشترک (ب.م.م) دو عدد m و n را محاسبه می‌کند.

۷. گزینه "ب" صحیح است.



```
int t;
```

```
t = strcmp("abcd","abcedb"); → -1
```

1- نشان می‌دهد که رشته دوم (abcedb) از رشته اول (abcd) بزرگتر است و اختلاف کاراکتر مورد نظر که در دو رشته باهم متفاوت است ۱ می‌باشد. یعنی کاراکتر e و کاراکتر d، به اندازه یک عدد باهم اختلاف دارند. لازم به ذکر است که کد اسکی کاراکتر e=101 و d=100 می‌باشد. (برای مشاهده لیست کدهای اسکی به پیوست ۲ مراجعه نمایید.)

نکته

در دو تابع strcmp() و strcmpi() ملاک مقایسه به هیچ وجه طول رشته‌ها نمی‌باشد.

تابع strcat() دو رشته را به هم متصل می‌کند و در انتهای آن کاراکتر 0 قرار می‌دهد.

مثال:

```
char str1[20]="Medical ", str2[]="Informatics";
strcat(str1,str2);
puts(str1); → Medical Informatics
```

نکته: تابع strcat()، محدوده رشته‌ها را بررسی نمی‌کند. لذا چنانچه طول رشته‌ی حاصل از الحاق دو رشته، بیشتر از گنجایش متغیر مقصد باشد، کاراکترهای اضافی در ادامه این متغیر در حافظه قرار می‌گیرند. این امر خطای کامپایلری ندارد ولی ممکن است منجر به از بین رفتن محتوای سایر خانه‌های حافظه شود.

تابع strset() یک کاراکتر را با تمام کاراکترهای رشته جایگذاری می‌کند.

مثال:

```
char str[]="Informatics";
strset(str,'*');
puts(str); → *****
```

نکته

می‌توان با تعریف یک آرایه دوبعدی از نوع کاراکتری، آرایه‌ای از رشته‌ها ایجاد کرد.



```
short int i,j,k;
for(i=0,j=9;i<10;i++,j--){
t=n[i]; n[i]=n[j]; n[j]=t;
}
```

- الف) عناصر آرایه متقارن می شوند. (ب) محتوای آرایه وارونه می شود.
ج) محتوای آرایه تغییر نمی کند. (د) جای نیمه سمت چپ و راست آرایه عوض می شود.

۶- با تعریف زیر چند بایت حافظه برای آرایه str در نظر گرفته می شود؟

```
char str[][6]={ "abc","fghi","lmnop"};
```

- الف) 15 (ب) 18 (ج) 20 (د) 12

۷- خروجی برنامه زیر چیست؟

```
char name1[]="SARA", name2[]="SARINA";
```

```
printf("%d",strcmp(name1,name2));
```

- الف) 1 (ب) 0 (ج) -8 (د) 8

۸- با تعریف زیر در زبان C، متغیر str چند بایت حافظه اشغال می کند؟

```
char str[][]={"Nima","Ali","Vahid"};
```

- الف) ۱۲ (ب) ۱۵ (ج) این تعریف خطا دارد. (د) ۱۸

۹- کدامیک از گزینه های زیر نادرست است؟

الف) `int a[2][3] = 0;` (ب) `int a[][3] = {{1,3},{6,2}};`

ج) `int a[2][3] = {{1,3,5},{4,6,2}};` (د) `int a[2][3] = {1,2,3,4,5,6};`

۱۰- در برنامه مقابل، اگر ورودی بصورت 45\01277 کد است؟

```
void main()
```

```
{
    char str[15];
    scanf("%s",str);
    printf("%s",str);
}
```

- الف) 45\01277 (ب) 45 (ج) 45 (د) 45277

۱۱- اگر در جواب برنامه زیر رشته ای با طول ۱۵ کاراکتر وارد شود، خروجی چه خواهد بود؟

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define N 20
```

```
int main()
```



```
{ {1,2,3}, {4,5,6} },
{ {7,8,9}, {10,11,12} },
{ {13,14,15}, {16,17,18} },
{ {19,20,0}, {0,0,0} }
};
```

مقدار `array[3][1][1]` با یک زیر خط نشان داده شده است.

۲. گزینه "د" صحیح است.

در هر بار اجرای حلقه، ابتدا شرط حلقه بررسی می‌شود و سپس محتوای خانه `i`ام که برابر با شماره اندیس یا همان `i` می‌باشد، تغییر می‌کند و دفعه بعد که حلقه اجرا می‌شود در واقع محتوای خانه `i+1`ام یعنی خانه بعد از `i` مورد بررسی قرار می‌گیرد که هنوز تغییر پیدا نکرده است. لذا بدیهی است که حلقه `while` تا زمانی که `i=20` نشده است (شرط `while`) برقرار است، ادامه پیدا خواهد کرد.

۳. گزینه "ج" صحیح است.

دقت داشته باشید که در انتهای حلقه `for` علامت `;` قرار دارد. در این حالت حلقه `for` بدنه‌ای نخواهد داشت. به عبارت دیگر حلقه `for` دستور اجرایی نخواهد داشت. حلقه ده بار اجرا می‌شود و هنگام خروج از حلقه مقدار `c=10` می‌باشد. کافیسست این مقدار را در عبارت `b[c] : b[c] - b[c] % 2 ? a += b[c]` قرار داده و نتیجه را که صفر است با مقدار قبلی `a` که آن هم صفر است جمع کنیم. در نهایت مقدار `a=0` خواهد بود. لذا مقداری که دستور `printf()` چاپ می‌کند عدد `0` خواهد بود.

اگر `;` جلوی حلقه `for` نمی‌بود خروجی برنامه 27 یعنی اختلاف مجموع اعداد زوج و فرد (بصورت اعداد فرد $\sum$ - اعداد زوج $\sum$) می‌شد.

۴. گزینه "ج" صحیح است.

حلقه در مجموع ۵ بار اجرا می‌شود و مقدار متغیرهای `a, b, c, d` در طول برنامه تغییری نمی‌کند. دستور `x[i] = (a, b, c, d, a+b+c+d);` عملگر `x[i]` که حاصل آخرین عبارت سمت راست یعنی `a+b+c+d` در متغیر مورد نظر ریخته شود. در نتیجه عدد 10 پنج بار در خروجی چاپ می‌شود. در انتهای برنامه محتوای تمام خانه‌های آرایه `x` عدد 10 خواهد بود.

۵. گزینه "ج" صحیح است.

این برنامه منجر می‌شود که جای عنصر اول آرایه با آخرین عنصر آن، عنصر دوم آرایه با عنصر یکی مانده به آخر و الی آخر جابجا شوند. اما توجه داشته باشید زمانی که این کار برای نیمه اول آرایه انجام می‌گیرد، تمام عناصر نیمه اول با آرایه با نیمه دوم آن جابجا شده‌اند و به عبارت دیگر آرایه معکوس شده است. اما زمانی که جابجایی عناصر همچنان تا انتهای آرایه ادامه پیدا کند. در این حالت عناصر نیمه دوم آرایه با عناصر نیمه اول آرایه جابجا خواهند شد. به عبارت