

آزمون میان‌دوره‌ای درس برنامه‌نویسی C/C++

شنبه ۲۵ تیرماه ۱۳۹۰

نصیری شرق، شاه‌محمدی

مدت آزمون: ۹۰ دقیقه

بخش یکم) کوتاه پاسخ‌های ضرور! ۱۵ نمره

به پرسش‌های زیر در یک یا حداکثر دو خط جواب مختصر، مفید و جامع بدهید.

۱. (۱ نمره) کاراکترهایی که در نامگذاری یک متغیر می‌توانند استفاده شوند کدامند؟ چه ترکیب‌هایی از این کاراکترها مجاز است؟
۲. (۲ نمره) Camel Naming و Dvorak چیستند؟ تفاوت‌ها و شباهت‌های این دو را بیان کنید.
۳. (۲ نمره) تفاوت بین یک Coder و یک Programmer چیست؟ در یک پروژه معمولاً کدام یک کارش را زودتر شروع می‌کند؟
۴. (۲ نمره) یک تعریف ساده از «هوشمندی» (Intelligence) ارائه دهید. چگونه یک یخچال می‌تواند هوشمند باشد؟
۵. (۲ نمره) ۴ زبان برنامه‌نویسی رایج و قوی را نام ببرید. کدام زبان از بقیه بهتر است؟ چرا؟
۶. (۳ نمره) سه کلمه‌ی using namespace std هر کدام به چه معنا هستند؟ مختصراً توضیح دهید.
۷. (۳ نمره) مفهوم «استفاده مجدد» (reusability) در طراحی Black Box ها به چه معناست؟ یک مثال از وجود این قابلیت در اتومبیل‌های سبک (نظیر پراید) و یک مثال از وجود این قابلیت در کدهایی که در هفته گذشته می‌توانسته‌اید زده باشید، بیان کنید.

پرسش دوم) بزن بریم کامپیوتر بخیریم! ۱۰ نمره

کامران می‌خواهد یک رایانه شخصی بخرد. در بازار او متوجه می‌شود که برخی سیستم‌های قابل خرید به نام ۳۲ بیتی و برخی ۶۴ بیتی هستند.

۱. (۲ نمره) این تفاوت به چه معناست؟
۲. (۲ نمره) دو قطعه سخت‌افزاری که بین این دو سیستم الزاماً متفاوتند و دو قطعه که مشترک هستند را نام ببرید.
۳. (۲ نمره) اگر کامران سیستم ۳۲ بیتی بخرد، در کدام قطعه سخت‌افزاری دیگر محدودیت وجود خواهد داشت؟ چرا؟
۴. (۲ نمره) بین یک پردازنده ۴ گیگا هرتزی ۶۴ بیتی و یک پردازنده ۵ گیگا هرتزی ۳۲ بیتی کدام برای اجرای برنامه‌های کامپیوتری که یک برنامه‌نویس ساده و مبتدی (دانش‌پژوه!) می‌نویسد سریع‌تر است؟ توضیح دهید.
۵. (۲ نمره) شغل این آقا کامران را X می‌نامیم. برای چه X هایی شما اکیداً خرید سیستم ۳۲ بیتی را توصیه می‌کنید؟ برای چه X هایی اکیداً ۶۴ بیتی؟ برای هر حالت یک مثال با ذکر دلیل بیان کنید.

پرسش سوم) کد ساده گلابی و هلو! ۲۰ نمره

از بین سه برنامه‌ی زیر دو تا را به دلخواه بنویسید. برنامه‌های شما باید کامل کامل، خوانا (آدمیزادی)، معقول، کوتاه و قابل کامپایل باشند!

۱. با استفاده از ۳ حلقه for و یک دستور if، برنامه‌ای بنویسید که تمام جایگشت‌های اعداد ۱ تا ۴ را چاپ کند.
۲. با استفاده از ۲ حلقه for و یک دستور if، برنامه‌ای بنویسید که تمام زیرمجموعه‌های مجموعه‌ی اعداد ۱ تا ۱۰ را چاپ کند.
۳. با استفاده از ۱ حلقه for و سه دستور if، برنامه‌ای بنویسید که سومین بزرگترین عدد در یک آرایه `int a[100]` حاوی ۱۰۰ عدد مثبت متفاوت را چاپ کند.

بخش چهارم) ای انسان، ای ماشین، دعوا نکنید! ۱۵ نمره

به پرسش‌های زیر در مورد مقایسه انسان و ماشین مختصر، مفید و جامع در حد دو خط الی یک پاراگراف (بسته به سؤال) پاسخ دهید.

۱. (۳ نمره) Captcha چیست؟ چرا انسان می‌تواند Captcha را راحت بخواند ولی برای ماشین سخت است؟ چرا ماشین می‌تواند

حاصل جمع ارقام 1390^{2011} را سریعاً حساب کند ولی انسان (بدون ماشین) نمی‌تواند؟

۲. (۲ نمره) بسیاری از مسائل چالش برانگیز بین انسان و ماشین (نظیر سؤال بالایی) مربوط به سرعت عمل است. و به‌وضوح اگر به

انسان هم فرصت مناسب (متناهی) داده شود می‌تواند حاصل جمع ارقام 1390^{2011} را سریعاً حساب کند. اما می‌دانیم رایانه‌های

کنونی هنوز در حل خیلی از مسائل از انسان عقب هستند. چه تیپ مسائلی توسط رایانه‌های کنونی به‌هیچ‌وجه قابل حل نیستند

ولی توسط انسان به‌سادگی قابل حل هستند؟ دو مثال بزنید.

۳. (۲ نمره) آیا مسئله‌ای (از تیپ ورودی/خروجی اطلاعات؛ و نه مثلاً برش یک تار مو به ۱۰۰۰ رشته موازی!) وجود دارد که انسان در

زمان خیلی خیلی زیاد (اما متناهی) نتواند آن را حل کند اما ماشین بتواند؟ اگر خیر چرا؟ اگر بلی دو مورد مثال بزنید.

۴. (۵ نمره) فرض کنید از شما خواسته شده است تا تعداد زیردرخت‌های هم‌بند یک گراف ۹۰ رأسی داده شده (شکلش روی یک

کاغذ A4 برای شما کشیده شده!) را پیدا کنید. نموداری برای مراحل مختلف پیدا کردن جواب این مسئله، از «درک مسئله» تا

«جواب خروجی» رسم کنید. مشخص کنید هر کدام از مؤلفه‌های OS, Compiler, C++, برنامه نویسی و نهایتاً «کد زبان ماشین»

در کجای نمودار شما قرار دارد.

۵. (۳ نمره) فرض کنید مسئله‌ای آیت‌م قبل (پیدا کردن تعداد زیردرخت‌های ...) عیناً و با همین ورودی، به دانش‌پژوهان دوره

تابستانه‌ی المپیاد کامپیوتر سال ۱۴۴۰ ه.ش. (۵۰ سال آینده!) داده می‌شود. بالطبع مدتی که طول می‌کشد آن دانش‌پژوه به جواب

نهایی برسد، کمتر از مدت زمانی است که امروز ما صرف می‌کنیم. این کاهش زمان رسیدن به جواب (از لحظه‌ی دریافت کاغذ

حاوی شکل گراف ۹۰ رأسی تا لحظه‌ی پیدا کردن عدد خروجی) بیشتر در کدام قسمت نمودار رسم شده توسط شما جلوه دارد؟

بخش پنجم) تابع جمع‌الوسط! ۲۰ نمره

تابع SumP قرارست دو اشاره‌گر به دو خانه از وسط یک آرایه از int را به‌عنوان آرگومان گرفته و حاصل جمع تمام مقادیر بین این دو خانه (با

احتساب خود خانه‌ها) را برگرداند. این تابع را (با افزودن خطوطی در محل مشخص شده در کد زیر و بدون تغییر در سایر قسمت‌های کد) با

زمان اجرای خطی در حالتی بنویسید که:

۱. (۱۰ نمره) شما مجاز به تعریف آرایه یا تابع جدیدی نمی‌باشید. الزامی هم ندارد P حتماً به اندیس کمتری نسبت به q اشاره کند.

۲. (۱۰ نمره) علاوه بر محدودیت‌های قسمت ۱، شما مجاز به استفاده از عملگرهای < یا <= یا > یا >= نیستید.

```
1. #include <iostream>
2. using namespace std;
3.
4. long long SumP(int *p, int *q) {
5.     // کد خود را در این قسمت بنویسید
6. }
7.
8. int main() {
9.     int a[10];
10. for (int i=0; i<10; i++)
11.     a[i] = 1000 + i;
12. cout << "sum is" << SumP(&a[2], &a[5]) << endl; // عدد باید چاپ شود
13. return 0;
14. }
```

بخش ششم) برنامه‌ی مدیریت چراغونی آقا نوید! ۲۰+۱۰ نمره



به‌مناسب نیمه شعبان، اهالی کوچی پرستوی هفت (قربانعلی‌زاده‌ی سابق) قصد دارند کوچه را با آویزان کردن تعدادی ریسه چراغانی کنند. برای شبیه‌سازی روشن و خاموش بودن چراغ‌های ریسه‌ها، آن‌ها از نوید (برنامه‌نویس محله!) خواسته‌اند که به‌شان کمک کند.

می‌دانیم هر ریسه بین ۶ تا ۶۰ لامپ دارد و تعداد لامپ‌های هر ریسه با بقیه ریسه‌ها ممکن است متفاوت باشد. تعداد ریسه‌ها (که آن را W می‌نامیم) هم عددی نامعلوم است.

◀ مأموریت شماره یک) نگه‌داری اطلاعات یک ریسه ----- (۱۰ نمره)

نوید ادعا می‌کند که اطلاعات هر ریسه (شامل تعداد لامپ‌ها و وضعیت هر لامپ) را می‌تواند دقیقاً در یک متغیر نگه‌داری کند. دقت کنید که نوید هنوز struct یا مفاهیم مشابه را نخوانده و اطلاعاتش در حد اطلاعاتی هست که در کلاس تدریس شده است.

۱. (۲ نمره) به نظر شما نوید این کار را چگونه انجام می‌دهد؟ توضیح دهید.

۲. (۴ نمره) قطعه کد زیر را برای انجام این کار تکمیل کنید.

```
1. int a[100];
2. int n;
3. X data; // به جای X تایپ متغیر مربوطه را بنویسید.
4. // کد شما برای تبدیل اطلاعات فوق به یک متغیر در این قسمت است.
```

۳. (۴ نمره) قطعه کدی بنویسید که با گرفتن متغیر data، مجدداً یک آرایه (حاوی اطلاعات) و یک n (تعداد لامپ‌ها) را پر کند. در واقع این کد معکوس عمل خواسته شده (از نظر ورودی/خروجی) در مرحله قبل را باید انجام دهد.

◀ مأموریت شماره دو) تغییر وضعیت ریسه‌ها ----- (۱۰ نمره)

پس از مأموریت یکم، نوید توانست اطلاعات W ریسه را در یک آرایه از تایپ X ذخیره کند. اکنون اهالی محل از نوید می‌خواهند که تابعی بنویسد که یک ریسه (یک متغیر از تایپ X) و یک بازه متوالی از لامپ‌های آن ریسه (نظیر $[u, v]$) و یک متغیر «امر» (به اسم c) بگیرد و:

- اگر متغیر امر برابر با صفر بود، همه لامپ‌های شماره i که $u \leq i \leq v$ از ریسه مربوطه، خاموش شوند.
- اگر متغیر امر برابر با یک بود، همه لامپ‌های شماره i که $u \leq i \leq v$ از ریسه مربوطه، روشن شوند.
- اگر متغیر امر برابر با دو بود، همه لامپ‌های شماره i که $u \leq i \leq v$ از ریسه مربوطه، تغییر وضعیت داده شوند (اگر خاموش بوده، روشن و اگر روشن بوده، خاموش بشود).

برای انجام این منظور، به آقا نوید کمک کرده و تابع زیر را تکمیل کنید:

```
1. X DoAction(X data, int u, int v, int c) {
2.     // ابتدا باید دستور خواسته شده انجام شود و سپس
3.     // مقدار جدید اطلاعات باید در قالب یک متغیر X برگردانده شود.
4. }
```

◀ مأموریت شماره سه) بهینه‌سازی مأموریت قبلی ----- (۱۰ نمره)

نوید مأموریت قبلی را با استفاده از یک حلقه انجام داد (که راه عادی بود). اما از آنجا که تعداد این دستورها زیاد بود، نوید احساس کرد که برنامه را می‌تواند سریع‌تر کند. در واقع در حالت کنونی، اگر حداکثر تعداد لامپ‌های یک ریسه (که گفته‌ایم ۶۰ است) را به‌صورت پارامتریک با T نمایش دهیم، برای اجرای K دستور در بدترین حالت به $O(KT)$ زمان احتیاج داریم و نوید قصد دارد این زمان را بهبود بخشد. برای این منظور نوید مجاز است یک پیش پردازش و یک حافظه عمومی (global) از $O(T)$ داشته باشد. بهترین ایده خود را فقط شرح دهید. «پایان»!