



بسمه تعالی

تاریخ : ۳ تیرماه ۱۴۰۲

آزمون پایانی درس مبانی کامپیوتر و برنامه‌سازی به زبان C

مدت زمان آزمون : ۱۳۰ دقیقه

نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

مدرس : یحیی پورسلطانی

همانا با یاد خدا دل‌ها آرام می‌گیرد

نام و نام خانوادگی :

شماره دانشجویی :

رشته تحصیلی :

شماره صندلی :

اینجانب با شماره دانشجویی بر اساس شماره‌ی صندلی اعلام

شده ، در جلسه‌ی آزمون حاضر شده و شماره‌ی صندلی خود را با مشخصات خود تطبیق دادم.

امضا

دانشجوی فرهیخته، با سلام

۱. لطفا مشخصات خودتان را با شماره صندلی اعلام شده تطبیق داده و عبارت کادر بالا را تکمیل کنید. عدم تکمیل عبارت کادر فوق به منزله غیبت

شما در آزمون و درج نمره صفر می‌باشد.

۲. پاسخگویی به سوالات بدون خط خوردگی ، واضح و خوانا باشد.

۳. پاسخ سوالات را منحصرأ در محل تعیین شده بنویسید.

۴. نام و نام خانوادگی و شماره دانشجویی خود را در بالای تمامی صفحات اوراق آزمون بنویسید.

۵. این آزمون از ۱۰۰ نمره است و پس از تصحیح، به میزان گفته شده برای آزمون مقیاس خواهد شد.

در جدول و کادرهای زیر مطلقاً چیزی ننویسید.

سوال	۱	۲	۳	۴	۵	۶
امتیاز						

نمره نهایی :

پس از تجدید نظر :

شماره دانشجویی

نام و نام خانوادگی

لطفا در جدول زیر چیزی ننویسید

	توضیحات
	نتیجه بررسی
<p>اینجانب برگه‌ی امتحانی خود را در تاریخ ملاحظه کرده و تحویل گرفتم. اینجانب نسبت به برگه‌ی خود اعتراض دیگری را نداشته و پس از تحویل آن، هیچ کدام از اعتراضات بعدی (حتی اگر وارد باشد) مورد بررسی قرار نخواهد گرفت.</p> <p>نام و نام خانوادگی</p> <p>امضا :</p>	تاییدیه

بخش اول : سوالات مفهومی (۲۰ امتیاز)

۱. برای هرکدام از گزاره‌های زیر تعیین کنید که کدام یک درست و کدام یک غلط هستند. در صورت غلط بودن هرکدام، علت آن را در بخش مشخص شده در یک سطر توضیح دهید و یا آن را اصلاح کنید (۱۰ امتیاز).

گزاره	درست	غلط	توضیح (در صورت نادرست بودن)
نسبت دادن یک pointer به یک pointer دیگر، همواره غلط است.			
نام آرایه، همانند یک اشاره‌گر قابل تغییر عمل می‌کند.			
رشته‌ها، اشاره‌گرهایی به کاراکتر اولشان هستند.			
با جمع کردن عدد n با یک pointer، حاصل نهایی به اندازه‌ی n بیت بیشتر خواهد شد.			
ارسال یک ساختار به تابع، به صورت پیش‌فرض به صورت Pass By Reference است.			
در صورت بازکردن یک فایل با تابع fopen، در صورتی که آن فایل وجود نداشته‌باشد، این تابع مقدار NULL را بر می‌گرداند (return می‌کند).			

۲. فرض کنید که تصویر زیر، بخشی از حافظه است و در کنار هر خانه، آدرس آن خانه را نوشته‌ایم (اعداد فرضی هستند).

متغیر x	۱۰۰۰	۲۰۰
متغیر y	۱۰۰۸	۱۰۰
متغیر sum	۱۰۱۶	

```

4  int main()
5  {
6      int x=200;
7      int y=100;
8      int sum;
9
10     int* xPtr = &x;
11     int* yPtr = &y;
12
13     sum = *xPtr+ *yPtr;
14     return 0;
15 }
```

نام و نام خانوادگی

شماره دانشجویی

اگر هرکدام از عبارات زیر را با دستور printf چاپ کنیم، چه مقداری در صفحه نشان داده می‌شود؟ ضمن نوشتن عبارت چاپی، علت چاپ آن را نیز توضیح دهید (۱۰ امتیاز).

عبارت	مقداری که این عبارت به ما می‌دهد	مفهوم عبارت
x		
*&x		
&*yPtr		
&y		
sum		

بخش دوم : تحلیل کد (۲۵ امتیاز)

خطاهای کد زیر را اعلام کرده و آن را اصلاح کنید (توجه کنید که اعلام خطاهای بی‌جا منجر به دریافت نمره نخواهد شد). این کد

```

5  int main()
6  {
7      char key[10] = "final@402#SUT";
8      char pass[10];
9      scanf("%s", &pass);
10     if(key==pass){
11         printf("Correct Password!\n");
12     }else{
13         printf("Wrong Password!\n");
14     }
15     return 0;
16 }
```

یک رمز را از کاربر دریافت کرده و با مقدار متغیر key مقایسه می‌کند. در صورتی که رمز وارد شده با مقدار متغیر key برابر باشد، عبارت Correct Password در صفحه چاپ می‌شود. در غیر این صورت، عبارت Wrong Password چاپ می‌شود.

(۵ امتیاز)

شماره دانشجویی

نام و نام خانوادگی

شماره خط	علت خطا	اصلاح خطا

۳. خروجی کدهای زیر را بیابید (هر کدام ۵ امتیاز دارد). توجه فرمایید که به توضیحات اضافی، نمره‌ای تعلق نخواهد گرفت و لازم است

صرفاً خروجی را در کادرهای مشخص شده بنویسید- در تمام سوالات فرض کنید کتابخانه‌های مورد نیاز را افزودیم (۲۰ امتیاز).

الف

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main()
6  {
7      int a=1;
8      int *p;
9      p=&a;
10     a++;
11     printf("%d", a*p);
12     return 0;
13 }
```

شماره دانشجویی

نام و نام خانوادگی

ب

```

5  int main()
6  {
7      int n=3;
8      int* p;
9      p=&n;
10     if(p!=&n) {
11         printf("OK\n");
12     }else{
13         printf("NOK\n");
14     }
15     return 0;
16 }
```

ج

```

5  int main()
6  {
7      int n=3;
8      int* p;
9      p=&n;
10     printf("%d\n", *p+n+ *&n);
11
12     return 0;
13 }
```

د

```

5  int main()
6  {
7      int a[] = {1,2,3,4,5,6,7,8};
8      int *p,*q;
9      p=a;
10     q=a+8;
11     do{
12         printf("%d\n", *p+*q);
13         p++;
14         q--;
15     }while(p!=q);
16 }
```

بخش سوم: خرده کدنویسی (۵۵ امتیاز)

۴. می‌خواهیم برنامه‌ای را بنویسیم که یک خط متنی را دریافت کند و پس از آن که لیستی را از کلمات آن ایجاد کرد (فرض می‌کنیم که کلمات با فاصله - Space - جدا شده‌اند)، یک تحلیل از فراوانی توزیع طول کلمات به ما بدهد. برای این منظور، کد ناقص زیر را در نظر بگیرید.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  //structs
6  typedef struct repeat{
7      int lessThan3; // length<3
8      int between3To5; // 3 <= length <5
9      int longerThan5; // 5 < length
10 } TextInfo;
11
12 //prototypes
13 void splitWords(char* , char[100][100], int*);
14 void extractInfo(TextInfo*, char[100][100] , int);
15 void printInfo(TextInfo textInfo);
16 |
17
18 int main()
19 {
20     char wordList[100][100];
21     TextInfo info = {0,0,0};
22     int numberOfWords=0;
23
24     char line[100];
25     gets(line);
26
27     splitWords(line, wordList, &numberOfWords);
28     extractInfo(&info, wordList, numberOfWords);
29     printInfo(info);
30
31 }
```

همان طور که مشاهده می کنید، این برنامه دارای سه تابع است که قرار است بخش های مهمی از آن را شما بنویسید! این توابع بدین شرح هستند :

✓ تابع Split Words : این تابع یک خط متنی را دریافت کرده و پس از جداسازی کلمات آن، آن کلمات را در آرایه ای دوبعدی word List ذخیره می کند و در نهایت، **تعداد کلمات** را در متغیر number Of Words ذخیره می کند. توجه کنید که این متغیر عددی در قالب اشاره گر به این تابع داده می شود و پس از محاسبه ی تعداد کلمات، لازم است مقدار آن را در محل (آدرس) داده شده ذخیره کنید.

✓ تابع extract Info : در این تابع، آدرس یک ساختار از نوع Text Info را به همراه فهرست کلمات و تعداد آن ها را بر اساس یک خط از متن می دهیم (اطلاعات مورد نیاز توسط تابع Split Words بدست آمده است) . در این تابع، کلمات بر اساس سائزشان (تعداد حروفشان) تقسیم بندی می شوند و در سه دسته ی کمتر از ۳ تا حرفی، بین ۳ تا ۵ حرفی و بیشتر از ۵ حرفی تقسیم می شوند. تعداد اعضای هر کدام از این سه دسته، در سه فیلد ساختار (به ترتیب با نام های less Than 3 و between 3 and 5 و در نهایت longer Than 5 ذخیره می شوند).

✓ تابع print Info اطلاعات ذخیره شده در ساختار را چاپ می کند.

مثال : یک خط متن زیر را در نظر بگیرید :

Alligator is a big long animal with sharp teeth.

برای این ورودی لازم است خروجی زیر تولید شود :

```
[0,3) ---> 2
[3,5) ---> 3
[5,INF) ---> 4
```

چرا که در آن خط از متن، دو کلمه داریم که بین ۰ تا ۳ حرفی هستند، سه کلمه داریم که بین ۳ الی ۵ حرف و چهار کلمه داریم که بیش از ۴ حرف طول دارند. ابتدای هر کدام از بازه ها بسته بوده و انتهایشان باز است.

نام و نام خانوادگی

شماره دانشجویی

الف) بر اساس توضیحات داده شده، تابع `split Words` را بنویسید. برای پیاده سازی این تابع، نباید از توابع آماده مثل `strtok` استفاده کنید (در صورت استفاده نمره ای تعلق نخواهد گرفت). برای راحتی شما، بخشی از آن را برایتان نوشتیم و بقیه ی آن را در کادرهای خالی بنویسید. (راهنمایی: در بدنه حلقه `do while` لازم است کلمات را بخوانید و پس از حلقه، تعداد کلمات را بدست آورده و ذخیره کنید) –

(۲۰ امتیاز)

```

33 void splitWords(char* line , char wordList[100][100], int* numberOfWordsPtr){
34     char currentChar = line[0];
35     int lineCharCounterIndex=0;
36     int wordCharCounterIndex=0;
37     int wordCounter=0;
38
39     do{
40         char word[100];
41         wordCharCounterIndex =0;
42         currentChar = line[lineCharCounterIndex];
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61     }while(currentChar!='\0');
62
63
64 }

```

آزمون پایانی (نظری) مبانی کامپیوتر و برنامه سازی به زبان C – گروه ۲

شماره دانشجویی

نام و نام خانوادگی

ب) تابع extract Info را مطابق با توضیحات پیاده سازی فرمایید (به پروتوتایپ تابع دقت کنید) – ۱۰ امتیاز

ج) تابع print Info را به گونه ای پیاده سازی کنید که خروجی را مطابق با تصویر صفحه ی ۸ چاپ نماید (۵ امتیاز).

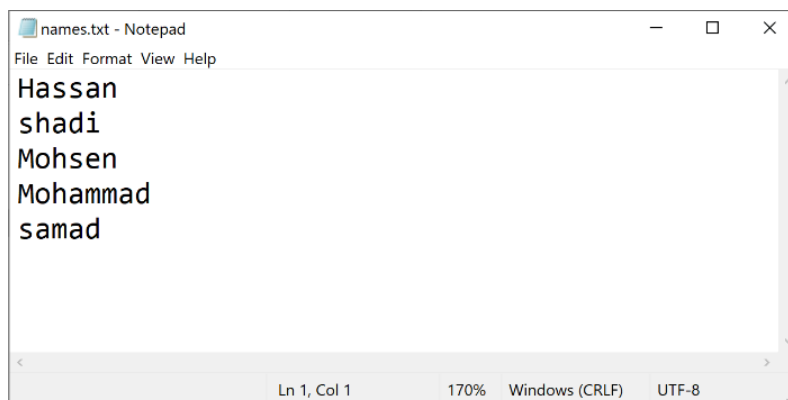
شماره دانشجویی

نام و نام خانوادگی

۵. می‌خواهیم برنامه‌ای را بنویسیم که یک کلید (کلمه) را در یک فایل متنی جستجو نماید. در فایل متنی مفروض، تعدادی اسم را به صورت تک به تک در هر خط نوشته‌ایم (در نتیجه حواسمان هست که هر خط با یک کاراکتر `\n` پایان می‌پذیرد). برای این منظور، برنامه‌ی زیر را نوشته‌ایم :

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int isExistInFile(char filePath[100], char key[10]);
5
6  int main()
7  {
8      char key[100];
9      char path[100];
10
11     printf("Enter File Path : ");
12     scanf("%s", path);
13
14     printf("Enter key : ");
15     scanf("%s", key);
16
17
18     int isExist= isExistInFile(path,key);
19
20     if(isExist==1){
21         printf("Key exists in %s\n", path);
22     }else if(isExist ==0){
23         printf("Key *NOT* exists in %s\n", path);
24     }else if(isExist ==-1){
25         printf("File Not exists in %s\n", path);
26     }
27
28     return 0;
29 }
```



شماره دانشجویی

نام و نام خانوادگی

در این برنامه، یک مسیر ذخیره‌ی فایل و یک کلید را دریافت می‌کنیم. سپس، این دو رشته را به تابع `is Exist In File` منتقل می‌کنیم. این تابع، یکی از سه مقدار زیر را بر می‌گرداند :

✓ مقدار ۱ در صورتی که کلید در فایل موجود باشد.

✓ مقدار صفر در صورتی که کلید در فایل موجود نباشد.

✓ مقدار ۱- در صورتی که فایلی که آدرس آن را به تابع داده‌ایم وجود نداشته‌باشد.

با ملاحظات گفته شده، تابع `is Exis In File` را پیاده‌سازی کنید. برای پیاده‌سازی آن، حتما به پروتوتایپ آن (واقع در خط ۴) توجه کرده و بر مبنای آن، پیاده‌سازی کنید. (راهنمایی : توجه کنید که در پایان هر کدام از نام‌ها، یک کاراکتر `\n` وجود دارد). (۲۰ امتیاز)

پيروز، شاد و تن درست باشيد