



باسمہ تعالیٰ

تاریخ : ۵ اردیبهشت ۱۴۰۱

مدت زمان آزمون : ۱۳۰ دقیقه آزمون میان ترم درس مبانی کامپیوتر و برنامه سازی به زبان C

مدرس : یحیی پورسلطانی

نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

همانا با یاد خدا دل ها آرام می گیرد

نام و نام خانوادگی :

شماره دانشجویی :

رشته تحصیلی :

شماره صندلی :

اینجانب با شماره دانشجویی بر اساس شماره ی صندلی اعلام

شده ، در جلسه ی آزمون حاضر شده و شماره ی 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 می باشد.

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 میان ترم مقیاس خواهد شد.

در جدول و کادرهای زیر مطلقا چیزی ننویسید.

سوال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
امتیاز										

نمره نهایی :

پس از تجدید نظر :

شماره دانشجویی

نام و نام خانوادگی

لطفا در جدول زیر چیزی ننویسید

	توضیحات
	نتیجه بررسی
	تاییدیه

نام و نام خانوادگی

شماره دانشجویی

بخش اول : سوالات مفهومی (۲۲ امتیاز)

۱. انواع زبان‌های برنامه‌سازی را از نظر نزدیکی به زبان انسان طبقه کرده و ضمن بیان یک ویژگی از هر کدام، برای هر کدام مثال بزنید (۹ امتیاز).

نوع زبان برنامه نویسی (طبقه)	ویژگی‌هایی که این دسته از زبان‌های برنامه نویسی دارند	مثال

۲. اساسی ترین نرم‌افزاری که در یک کامپیوتر باید وجود داشته‌باشد تا امکان اجرای برنامه‌های دیگر فراهم شود کدام است (۱ امتیاز)?

.....

۳. زبان‌های برنامه‌نویسی مفسری و کامپایلری چه تفاوت‌هایی با یک دیگر دارند؟ دو مورد را ذکر کرده و هر کدام را در دو خط شرح دهید (۴ امتیاز).

۱-

.....

۲-

.....

نام و نام خانوادگی

شماره دانشجویی

۴. در خصوص حافظه‌ی اصلی و حافظه‌ی ثانویه، گزینه‌های درست را علامت بزنید (۴ امتیاز).

سوال	حافظه‌ی اصلی	حافظه‌ی ثانویه
حافظه‌ی RAM از این نوع است.		
هارد دیسک از این نوع است.		
با خاموش شدن کامپیوتر، اطلاعات در این نوع از حافظه از بین می‌رود.		
پس از ذخیره‌سازی اطلاعات، در این نوع از حافظه ذخیره می‌شود.		
در زمان اجرای برنامه، اطلاعات آن برنامه در این نوع از حافظه بارگزاری می‌شود.		

۵. انواع دستورات اصلی برنامه‌نویسی را در زبان‌های برنامه نویسی نام برده و بگویید که در زبان C چه ساختارهایی برای پیاده‌سازی این

دستورات ارائه شده است (۴ امتیاز).

نوع دستور	ساختاری که در زبان C برای پیاده‌سازی این نوع از دستورات پیشبینی شده است.
دستورات دنباله‌ای (sequential)	به صورت پیش فرض، دستورات به صورت خط به خط و به ترتیب اجرا می‌شوند.

بخش دوم: تحلیل کد (۱۸ امتیاز)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      |
7      |
8      return 0;
9  }
10

```

۶ در هرکدام از موارد زیر، خروجی کد داده شده را در کادر مشخص شده، بنویسید (صرفاً

خروجی را بنویسید- به توضیحات بیشتر نمره‌ای تعلق نمی‌گیرد). در پاسخگویی به

سوالات فرض کنید که قطعه کد داده شده، با کادر مشکی کد مقابل، جایگزین خواهد

شد (۱۲ امتیاز).

الف

```

int a[10];
int sum=0;
for(int i=0; i<10; i++)
    a[i]=i;
for(int i=0; i<9; i++)
    a[i]+=a[i+1];
for(int i=0; i<10; i++)
    sum+=a[i];
printf("%d", sum);

```

ب

```

int x=9;
int y=2;
float a=9;
float b=2;
printf("%f", x/y + a/b + x/b);

```

ج

```

int x=100;
for(int i=10; i<=30; i+=3)
    if(i%2==0)
        x-=i;
printf("%d", x);

```

د

```

int x;
int count=0;
for(x=-1; x<=10; x++){
    if(x<5)
        continue;
    else
        break;
    count++;
}
printf("%d\n", count);

```

شماره دانشجویی

نام و نام خانوادگی

۷. در هر کدام از موارد زیر، خطاهای خواسته شده را بدست آورید (۶ امتیاز).

الف) در قطعه کد زیر، خطاهای کامپایلری را بدست آورید و سپس آن را اصلاح کنید؛ توجه فرمایید که در صورت اعلام جابجا و یا ناصحیح خطاهای کامپایلری، به خطای اعلام شده نمره‌ای تعلق نمی‌گیرد (۳ امتیاز).

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     float i=2;
7     For(i=0; i<10; i+=1){
8         a +=i;
9         return i;
10 }
```

ب) تمایل داریم تا برنامه‌ای را بنویسیم که اعداد ۱ الی ۱۰۰ را باهم جمع کند و برای این منظور، برنامه‌ی زیر را نوشتیم. آیا این برنامه این خواسته

را برآورده می‌کند؟ در صورتی که پاسخ شما منفی است، پس از بیان علت منفی بودن پاسخ، راهکاری را برای اصلاح آن ارائه

دهید (۳ امتیاز)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 |
4 | =int main(){
5 |     int z=100;
6 |     int sum=0;
7 | - while(z>0){
8 |         sum+=z;
9 | -     }
10 |     printf("Sum = %d\n", sum);
11 | }
12 |
```

بخش سوم : خرده کد نویسی (۶۰ امتیاز)

۸. می‌خواهیم برنامه‌ای را بنویسیم که یک عدد ۵ رقمی را دریافت کرده و تشخیص دهد که آیا ارقام آن متقارن است یا خیر؟ به عنوان مثال، عدد ۱۲۳۴۵ متقارن نیست؛ اما عدد ۱۲۳۲۱ متقارن است؛ چرا که اگر از راست به چپ و یا از چپ به راست خوانده شود، یک مقدار دارد. برای حل این سوال می‌توان برنامه‌ای را نوشت که **معکوس یک عدد را با خود آن عدد** مقایسه کند. بخشی از این برنامه را برای شما نوشته‌ایم (کد زیر را ببینید). لازم است شما به جای بخش مشخص شده (خط ۲۲ و به جای کامنت نوشته شده با عبارت Your Code) یک کدی را بنویسید که معکوس عدد را محاسبه کرده و در reverse ذخیره کند. کد خود را در کادر زیر بنویسید (۱۰ امتیاز).

راهنمایی : یک عدد را می‌توان به صورت مجموعی از توان‌های ۱۰ نوشت. همچنین اجازه استفاده از تابع pow(a,b) را برای محاسبه توان دارید.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  int isPalindrome5Digit(int number);
5
6  int main()
7  {
8      int number=0;
9      scanf("%d", &number);
10     if(isPalindrom5Digit(number)){
11         printf("The Number is Palindrome\n");
12     }else{
13         printf("The Number is NOT Palindrome\n");
14     }
15     return 0;
16 }
17
18 int isPalindrome5Digit(int number){
19     if(number>10000 && 99999>number){
20         int reversed=0;
21
22         //Your Code!
23
24         if(number==reversed){
25             return 1;
26         }else{
27             return 0;
28         }
29     }else{
30         return 0;
31     }
32 }
33
34 }
```

شماره دانشجویی

نام و نام خانوادگی

۹. همان طور که به خاطر دارید، n امین جمله ی دنباله ی فیبوناچی به صورت زیر تعریف می شد :

$$f_n = \begin{cases} 1 & \text{if } n = 0 \text{ or } 1 \\ f_{n-1} + f_{n-2} & \text{otherwise} \end{cases}$$

فاکتوریل فیبوناچی به ازای n به صورت زیر تعریف می شود:

$$F_n = \prod_{k=0}^n f_k = f_0 f_1 \dots f_n$$

در ادامه، برنامه ای را می خواهیم بنوسیم که فاکتوریل فیبوناچی را به صورت بازگشتی برای ما بدست آورد. این برنامه به صورت زیر نوشته می شود:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int fibo(int a);
4  int fiboFact(int a);
5  int main(){
6      int inputNumber=0;
7      scanf("%d", &inputNumber);
8      printf("%d", fiboFact(inputNumber));
9  }
10
11 int fibo(int a){
12     //PART A
13 }
14
15 int fiboFact(int a){
16     //PART B
17 }
```


نام و نام خانوادگی

شماره دانشجویی

الف (در ابتدای کار لازم است یک تابع به منظور بدست آوردن n امین جمله دنباله فیبوناچی بنویسید. کد این تابع را به صورت بازگشتی در کادر زیر بنویسید. فرض کنید که کد شما قرار است در بدنه تابع fibo و به جای کامنت PART A در شکل قبل جایگذاری شود (۱۰ امتیاز).

ب (در نهایت، با استفاده از تابعی که در بخش الف نوشته‌اید و بر اساس تعریف فاکتوریل دنباله‌ی فیبوناچی، کد یک تابع بازگشتی را به منظور محاسبه‌ی فاکتوریل دنباله‌ی فیبوناچی در کادر زیر بنویسید. فرض کنید که کد شما در بدنه‌ی تابع fiboFact و به جای کامنت PART B جایگذاری می‌شود (۱۰ امتیاز).

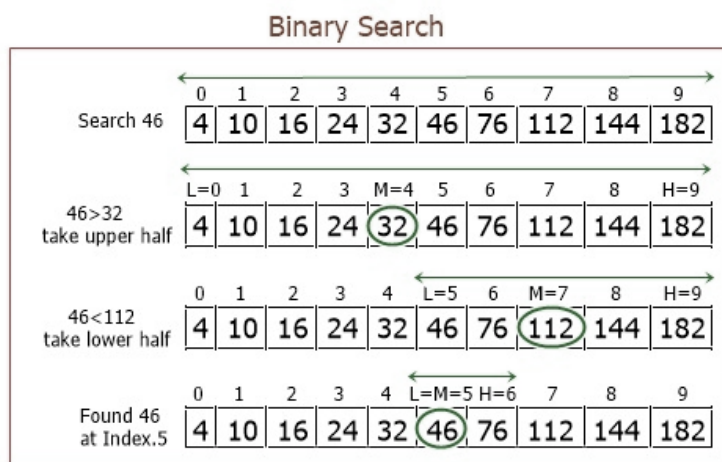
۱۰. آرایه‌ای با n عنصر و به صورت مرتب شده مفروض است. هدف، نوشتن تابعی است که یک عنصر داده شده مانند x را در آرایه مذکور جستجو کند. یک رویکرد ساده برای انجام این کار، استفاده از جستجوی خطی (Linear Search) است؛ اما به خاطر داریم که در جستجوی خطی، همه عناصر آرایه با x مقایسه می‌شوند. برای آن که مجبور به بررسی همه‌ی عناصر آرایه نشویم، از رویکرد دیگری به نام جستجوی دودویی (Binary Search) استفاده می‌کنیم. ایده اصلی نهفته در پس جستجوی دودویی، استفاده از اطلاعات موجود در یک آرایه‌ی از قبل مرتب شده است. در جستجوی دودویی اساساً نیمی از عناصر، تنها پس از یک مقایسه حذف می‌شوند. برای انجام جستجو برای یافتن یک عنصر مثل x ، از الگوریتم زیر استفاده می‌شود.

۱. x با عنصر میانی آرایه مقایسه می‌شود.

۲. اگر x با عنصر میانی آرایه یکی بود، اندیس عنصر میانی را بازگردان.

۳. در غیر این صورت، اگر x بزرگ‌تر از عنصر میانی بود، امکان دارد x در نیمه سمت راست آرایه و پس از عنصر میانی قرار داشته باشد (شایان توجه است که همانطور که پیش‌تر اشاره شد، آرایه مرتب شده است. پس در این حالت، نیمه‌ای با مقادیر بزرگ‌تر برای ادامه جستجو گزینش می‌شود).

۴. در غیر این صورت، اگر x از عنصر میانی آرایه کوچک‌تر باشد، آرایه به دو نیمه شکسته شده و جستجو در نیمه سمت چپ (با مقادیر کوچک‌تر از میانه)، ادامه پیدا می‌کند.



شماره دانشجویی

نام و نام خانوادگی

برای درک بهتر، به مثال بالا توجه کنید: در مثال قبل، یک آرایه به صورت مرتب شده داده شده است و هدف ما، پیدا کردن اندیس عنصر ۴۶ است. بنابراین، از عنصر ۰ الی عنصر ۹ را جستجو می‌کنیم. برای این منظور، میانه‌ی این بازه را محاسبه می‌کنیم (در صورتی که تعداد عناصر زوج نباشد، عنصر سمت چپ میانه را به عنوان میانه در نظر می‌گیریم). سپس میانه را با ۴۶ مقایسه می‌کنیم و اگر:

۱. عنصر میانه با ۴۶ برابر بود، حضور آن را اعلام می‌کنیم (اندیس آن را بر می‌گردانیم).

۲. اگر عنصر میانه از ۴۶ کمتر بود، سمت راست آرایه مورد جستجو قرار می‌گیرد (برای این منظور، کران سمت چپ جستجو یک واحد جلوتر از میانه قرار می‌گیرد).

۳. اگر عنصر میانه از ۴۶ بیشتر بود، سمت چپ آرایه مورد جستجو قرار می‌گیرد (برای این منظور، کران راست چپ جستجو یک واحد عقب‌تر از میانه قرار می‌گیرد).

در صورتی که کران سمت راست آرایه، عقب‌تر از کران سمت چپ آرایه بیفتد، به معنای آن است که **عنصر مورد نظر یافت**

نشده است و تابع مورد نظر، ۱- را بر می‌گرداند.

```

1  #include<stdio.h>
2  int main(void)
3  {
4      int arr[] = { 2, 3, 4, 10, 40 };
5      int n = sizeof(arr) / sizeof(arr[0]);
6      int x = 10;
7      int result = binarySearch(arr, 0, n - 1, x);
8      (result == -1) ? printf("Element is not present"
9                          " in array")
10                     : printf("Element is present at "
11                             "index %d",
12                             result);
13      return 0;
14  }
15
16  int binarySearch(int arr[], int leftBound, int rightBound, int x)
17  {
18      while (/*شرط توقف جستجو*/) {
19
20          /**
21           در اینجا باید در بازه مناسب به دنبال
22           عنصر بگردید و اگر پیدا نکردید، بازه را اصلاح کرده
23           و در صورتی که یافتید، اندیس آن را برگردانید.
24           */
25
26      }
27
28      return -1;
29  }
30

```

شماره دانشجویی

نام و نام خانوادگی

الف) با توجه به توضیحات ارائه شده، از شما می‌خواهیم که کد بدنه‌ی تابع `binary Search` را در کادر زیر بنویسید. برای این منظور لازم است که شرط توقف جستجو را در حلقه‌ی `while` بنویسید و سپس در حلقه‌ی `while`، بر اساس توضیحات ارائه شده، جستجو را انجام دهید (۱۵ امتیاز).

ب) بدون تغییر ورودی‌های تابع `binary search` آن را به یک تابع بازگشتی تبدیل کنید تا جستجو را بدون نیاز به حلقه `while` و با استفاده از بازگشت انجام دهد (۱۵ امتیاز).

پروژه شاد و تن‌دست‌باشد