

برنام خدا

آزمون مبانی برنامه نویسی ، زمان ۱۵۰ دقیقه

لطفاً روی برگه‌ی آزمون، پاسخنامه و دیگر برگه‌ها نام خود را بنویسید. لطفاً همه‌ی برگه‌های خود را پس از آزمون تحویل دهید. پاسخ‌گویی به پرسش‌ها در ۲۰ دقیقه‌ی نخست آزمون و سپس پس از هر نیم ساعت به مدت ۱۰ دقیقه خواهد بود ۴۵ دقیقه آخر آزمون به پرسشی پاسخ داده نمی‌شود. در حل مسأله‌های آزمون هیچ تابع ریاضی یا رشته‌ای آماده‌ای را به کار نبرید و تابع‌هایی را که نیاز دارید خودتان بنویسید

۱. خروجی برنامه زیر را بنویسید.(۱)

```

1 #include<stdio.h>          16 if(a || b)                  31 pi1=iap;
2 int main(){                 17 if(a && b)                  32 pi2=iap+3;
3 int a=1 , b=0, i=0;         18 if(a != b)i=1;            33 pi1=pi2;
4 char aP[]=                  19 else i=2;                  34 printf("%d  ",
5 {'a','d','g','y',          20 else i=3;                  35 *pi2);
6 'u','o','j','\0'          21 else i=4;                  36 pi1=++pi2;
7 };                           22 printf("%d  ",i);          37 printf("%d  ",
8 int iap[]=                  23 printf("%d  ",            38 ++*++pi1);
9 {10,20,30,40,50,          24 (i=++a-b--)+2);          39 pcl=aP;
10 60,70,80,90,100          25 a=12;                     40 printf("%c\n",
11 };                           26 printf("%d\n",            41 pcl[3]);
12 int *pi1,*pi2;              27 iap[i]);                  42 pcl+=5;
13 char *pcl;                  28 pi1=&a;pi2=&b;              43 printf("%s\n",pcl);
14 printf("%d\n",              29 pi1=pi2;                  44 return 0;
15 a? 12:48);                  30 printf("%d  ",a);         45 }

```

::solution::

```

12
3  4  30
12 40 61 y
oj

```

۲. تابع مقایسه‌ی دو رشته را به صورت زیر بنویسید که دو رشته را دریافت کند و اگر یکی (str1) کوچکتر از دومی (str2) بود مقدار منفی یک را برگرداند و اگر یکی بزرگ‌تر از دومی بود مثبت یک را برگرداند و اگر دو رشته طول مساوی و نویسه‌های (کاراکترهای) یکسان داشتند آن‌گاه صفر را برگرداند.(۱/۲۵)

"Ab" < "Abc" = "Abc" < "abc" < "bac" = "bac" < "bacd" < "da" = "da"

```
int str_cmp(const char *str1 , const char *str2 );
```

::solution::

```

1 #include<stdio.h>
2 int str_cmp(const char*str1 , const char*str2 );
3 int str_cmp2(const char*str1 , const char*str2 );
4 int main(){
5 int i;
6 char *st1[10]={ "abc","abc" , "abcd" , "abc" , "abc " ,"abc" };
7 char *st2[10]={ "Abc","abcd" , "abc" , "abc " , "abc " ,"abc" };
8 for(i=0;i<6;i++)
9 printf("%d \t %s \t\t %s\n",str_cmp(st1[i],st2[i]),st1[i],st2[i]);
10 for(i=0;i<6;i++)
11 printf("%d \t %s \t\t %s\n",str_cmp2(st1[i],st2[i]),st1[i],st2[i]);
12 return 0;
13 }
14 int str_cmp(const char*str1 , const char*str2 ){
15 int i;int retVal=0;
16 for(i=0;str1[i] && str2[i]; i++)

```

```

17     if (str1[i] < str2[i])
18     { retVal = -1; break; }
19     else if (str1[i] > str2[i])
20     { retVal = 1; break; }
21     if (retVal == 0) {
22         if (str1[i])
23             retVal = 1;
24         else if (str2[i])
25             retVal = -1;
26     }
27     return retVal;
28 }
29 int str_cmp2(const char*str1 , const char*str2){
30     int retVal=0;
31     for (;*str1 && *str2 ; str1++, str2++)
32         if (*str1 < *str2)
33             { retVal=-1; break; }
34         else if (*str1 > *str2)
35             { retVal=1; break; }
36     if (retVal == 0) {
37         if (*str1)
38             retVal = 1;
39         else if (*str2)
40             retVal = -1;
41     }
42     return retVal;
43 }

```

۳. محاسبه ترکیب m و n را به صورت بازگشتی می‌توان به صورت زیر تعریف کرد. تابعی به صورت بازگشتی بنویسید که دو عدد صحیح m و n را بگیرد و بر پایه‌ی فرمول داده شده ترکیب m و n را حساب کند. فرض کنید همواره $0 \leq m \leq n$ است. (۱/۲۵)

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1} \quad , \quad \binom{n}{0} = 1 \quad , \quad \binom{n}{n} = 1$$

```
int cmb(int n, int m);
```

::solution::

```

1 #include <stdio.h>
2 int cmb(int , int);
3 int main(){
4     int m,n;
5     printf("Enter n > 0 : "); scanf("%d",&n);
6     printf("Enter m >= n: "); scanf("%d",&m);
7     printf("%d\n",cmb(n,m));
8     return 0;
9 }
10 int cmb(int n, int m){
11     if(n==m || m==0) return 1;
12     return cmb(n-1,m)+cmb(n-1,m-1);
13 }

```

۴. تابع جستجوی دودویی را بنویسید که یک آرایه‌ی مرتب شده و غیر تکراری (a) را به همراه تعداد عنصرهای آن (n) و عددی که باید جستجو شود (v) را می‌گیرد و مکان v را در آرایه برمی‌گرداند و اگر نیافت منفی یک را برمی‌گرداند. (۱)

```
int bin_search(int a[], int n, int v);
```

::solution::

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

```

2 int bin_search(int a[], int n, int v);
3 int main(){
4     int v;
5     int a[]=
6         {0, 2, 7, 9, 12, 14, 20, 25, 32, 44} ;
7     printf("Enter number "); scanf("%d",&v);
8     printf("%d",bin_search( a, sizeof(a)/sizeof(int), v));
9     return 0;
10 }
11
12 int bin_search(int a[], int n, int v){
13     int end=n, start=0, mid;
14     while(start <= end){
15         mid=(start+end)/2;
16         if(a[mid] < v)      start      =mid +1 ;
17         else if(a[mid] > v) end      =mid -1;
18         else                break;
19     }
20     if(start>end) mid=-1;
21     return mid;
22 }

```

۵. تابع $\exp(x)$ را به کمک بسط سری $\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$ بنویسید. این تابع یک عدد اعشاری x را می‌گیرد و یک عدد اعشاری برمی‌گرداند که برابر e^x است. محاسبه سری را تا جایی ادامه دهید که آخرین جمله آن $\frac{x^n}{n!}$ کوچکتر از 0.001 شود. (۱/۷۵)

```

double exp(double x);

::solution::
1 #include<stdio.h>
2 double exp(double x);
3 int main(){
4     double x;
5     printf("Enter number "); scanf("%lf",&x);
6     printf("%lf",exp(x));
7     return 0;
8 }
9 double exp(double x){
10     double sum=x+1, term=x, nm=x, f=1; int i;
11     for(i=2; term>0.001; sum+=term, i++)
12         term = (nm *= x)/(f *= i);
13     return sum;
14 }

```

۶. تابعی بنویسید که اشاره‌گرهایی به دو عدد بزرگ را دریافت کند و سپس این دو عدد بزرگ را با هم جمع کند و نتیجه را برگرداند. ساختار عدد صحیح بزرگ در زیر نوشته شده است. این عددها بدون علامت هستند و همواره مثبت هستند. اگر به شکل دیگری نیز می‌توانید تابع جمع دو عدد صحیح بزرگ را بنویسید. (۱/۷۵)

```

typedef struct {int a[1000];int n;} hn; hn add(hn*a1, hn*a2);

::solution::
1 #include<stdio.h>
2 typedef struct {int a[1000];int n;} hn;
3 hn add(hn*a1, hn*a2);
4 int main(){
5     double x;

```

```
6  printf("Enter number "); scanf("%lf",&x);
7  printf("%lf",exp(x));
8  return 0;
9  }
10 hn add(hn*a1, hn*a2){
11     hn res;
12     return res;
13 }
```

تندرست باشید، احمد یوسفان