C第7次(第7周)作业（参考答案）

考试形式：开卷　　考试时间：2024-4-15

院系：东吴学院　年级：2023　专业：非计算机专业

学号：　姓名：　分数：

-、选择题（每小题2.0分，共20.0分）

01.A	02.D	03.A	04.A	05.D
06.A	07.B	08.D	09.A	10.B

二、填空题（每空2.0分，共20.0分）

01. 8
02. 9
03. 2
04. 3,5
    3,5
05. 4
06. fdf+abc=defdfd
07. 8
    50
    50,9

三、编程题（每小题6.0分，共60.0分）

01.(6.0分)答：

```
#include <stdio.h>
#include <string.h>
void fun(char  *s, char  t[])
{
    int i,j=0;
    for(i=0;s[i]!='\0';i++)
       if(!(i%2==0 && s[i]%2!=0))
         t[j++]=s[i];
    t[j]='\0';
}
int main()
{
    char   s[100], t[100];
    scanf("%s", s);
    fun(s, t);
    printf("%s", t);
}
```

02.(6.0分)答：

```
#include <stdio.h>
#include <math.h>
double fun(double  x[9])
{
   double sum=0;
   int i;
   for(i=0;i<8;i++)
      sum+=sqrt((x[i]+x[i+1])/2.0);
    return sum;
}
int main()
{
  double  s,a[9];
  int  i;
  for(i=0;i<9;i++)
    scanf("%lf",&a[i]);
  s=fun(a);
  printf("s=%f",s);
}
```

03.(6.0分)答：

```
#include <stdio.h>
#include <string.h>
void fun(char p1[], char p2[])
{
    int i,j;
    //寻找p1的结尾
    i=0;
    while(p1[i]!='\0') i++;
    //将p2中的字符依次复制到p1的末尾
    j=0;
    while(p2[j]!='\0')
    {
 p1[i]=p2[j];
 i++;
 j++;
    }
    //为p1添加字符串结束标记
    p1[i]='\0';
}
int main()
{
    char s1[80], s2[40];
```

```
    scanf("%s%s", s1, s2) ;
    fun(s1, s2);
    printf("%s", s1) ;
}
```

04.(6.0分)答：

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define M 100
void read(int* a, int m, int n)
{
    int i, j;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", a + i * n + j);
}
void write(int* a, int m, int n)
{
    int i, j;
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
            printf("%d ", *(a + i * n + j));
        printf("\n");
    }
}
int main()
{
    int a[M][M], m, n;
    scanf("%d%d", &m, &n);
    read(&a[0][0], m, n);
    write(&a[0][0], m, n);
    return 0;
}
```

05.(6.0分)答：

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
//在升序数组中查找，找到返回数组下标，失败返回-1
int binary_search(int key, int a[], int n)
{
    int low, high, mid, count = 0, count1 = 0;
    low = 0;
    high = n - 1;
    while (low <= high)              //查找范围不为0时执行循环体语句
```

```
    {
        count++;                    //count记录查找次数
        mid = (low + high) / 2;     //求中间位置
        if (key < a[mid])           //key小于中间值时
            high = mid - 1;         //确定左子表范围
        else if (key > a[mid])      //key 大于中间值时
            low = mid + 1;          //确定右子表范围
        else if (key == a[mid])     //当key等于中间值时，证明查找成功
        {
            //          printf("查找成功!\n 查找 %d 次!a[%d]=%d", count, mid, key);    //输出查找次数及所查找元素在数组中的位置
            //          count1++;               //count1记录查找成功次数
            //          break;
            return mid;
        }
    }
    return -1;
}
#define N 10
int main()
{
    int a[N], x, i;
    for (i = 0; i < N; i++)
        scanf("%d", &a[i]);
    scanf("%d", &x);
    if ((i = binary_search(x, a, N)) >= 0)
        printf("%d", i);
    else
        printf("无");
    return 0;
}
```

06.(6.0分)答：

```
#include <stdio.h>
#define  M  4
#define  N  5
int  fun ( int a[M][N] )
{
    int i,j,sum=0;
    for(i=0;i<M;i++)
      for(j=0;j<N;j++)
        if(i==0 || i==M-1 || j==0 || j==N-1)
          sum+=a[i][j];
    return sum;
```

```
}
int main( )
{  int  aa[M][N];
   int  i, j, y;
   for ( i=0; i<M; i++ )
      for ( j =0; j<N; j++ )
     scanf( "%d", &aa[i][j] );
   y = fun(aa);
   printf( "%d", y);
}
```

07.(6.0分)答：

```
#include <stdio.h>
void fun(int *s, int t, int *k)
{
    int i;
    *k=0;
    for(i=0;i<t;i++)
      if(s[i]>s[*k])
  *k=i;
}
int main( )
{
    int a[10], k;
    for(k=0;k<10;k++)
      scanf("%d",&a[k]);
    fun(a, 10, &k) ;
    printf("%d,%d", k, a[k]) ;
}
```

08.(6.0分)答：

```
#include <stdio.h>
#include <string.h>
#define   N   16
typedef  struct
{  char  num[10];
   int   s;
} STREC;
STREC  fun( STREC  *a, char *b )
{
  int i;
  STREC t = {"", -1};
  //在此添加代码
  for(i=0;i<N;i++)
  {
```

```
    if(strcmp(a[i].num,b)==0)
        return a[i];
  }
  return t;
}
int main()
{
 STREC  s[N]={{"GA005",85},{"GA003",76},{"GA002",69},{"GA004",85},
  {"GA001",91},{"GA007",72},{"GA008",64},{"GA006",87},
  {"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},
  {"GA011",77},{"GA017",64},{"GA018",64},{"GA016",72}};
   STREC  h;
   char  m[10];
   int  i;
   gets(m);
   h=fun( s,m );
   printf("%d",h.s);
}
```

09.(6.0分)答：

```
#include <stdio.h>
#include <string.h>
char* reverse(char s[])
{
 int i,l = strlen(s);
 char t;
 for (i = 0; i < l / 2; i++)
 {
  t = s[i];
  s[i] = s[l - 1 - i];
  s[l - 1 - i] = t;
 }
 return s;
}
char* convert(int x, char s[])
{
 int i = 0;
 while (x)
 {
  s[i] = x % 8;
  s[i] += '0';
  x /= 8;
  i++;
 }
```

```
 s[i] = 0;
 reverse(s);
 return s;
}
int main()
{
 int x;
 char s[20];
 scanf("%d", &x);
 convert(x, s);
 printf("%s", s);
 return 0;
}
```

10.(6.0分)答：

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
//在升序数组中查找，找到返回数组下标，失败返回-1
int binary_search(int key, int a[], int n)
{
    int low, high, mid, count = 0, count1 = 0;
    low = 0;
    high = n - 1;
    while (low <= high)              //查找范围不为0时执行循环体语句
    {
        count++;                     //count记录查找次数
        mid = (low + high) / 2;      //求中间位置
        if (key < a[mid])            //key小于中间值时
            high = mid - 1;          //确定左子表范围
        else if (key > a[mid])       //key 大于中间值时
            low = mid + 1;           //确定右子表范围
        else if (key == a[mid])      //当key等于中间值时，证明查找成功
        {
            //           printf("查找成功!\n 查找 %d 次!a[%d]=%d", count, mid, key);    //输出查找次
数及所查找元素在数组中的位置
            //           count1++;               //count1记录查找成功次数
            //           break;
            return mid;
        }
    }
    return -1;
}
#define N 10
int main()
```

```
{
    int a[N], x, i;
    for (i = 0; i < N; i++)
        scanf("%d", &a[i]);
    scanf("%d", &x);
    if ((i = binary_search(x, a, N)) >= 0)
        printf("%d", i);
    else
        printf("无");
    return 0;
}
```