

中山大学本科生期末考试

考试科目：《程序设计实验》(B 卷)

学年学期：2021–2022 学年第 1 学期

姓 名：_____

学 院/系：智能工程学院

学 号：_____

考试方式：闭卷

年级专业：_____

考试时长：120 分钟

班 别：_____

警示

《中山大学授予学士学位工作细则》第八条：“考试作弊者，不授予学士学位。”

以下为试题区域，共 4 道大题，总分 100 分，考生请在答题纸上作答

一、请由键盘读入正整数 N ，输出 $0 - N$ (包括 N) 之间的所有素数，按每行 5 个的形式输出。[要求：输入、输出要有提示]

输入示例：

请输入一个正整数：100

输出示例：

0-100 (包括100) 之间的所有素数有：

2	3	5	7	11
13	17	19	23	29
31	37	41	43	47
53	59	61	67	71
73	79	83	89	97

图 1 第一题

解：

参考代码：

```
1  #include<stdio.h>
2  int main()
3  {
4      int N, i, j, cnt = 0;
5      printf("请输入一个正整数:");
6      scanf("%d", &N);
7      printf("0- %d (包括 %d)之间的所有素数有:", N, N);
```

```

8      for (i = 2; i <= N; i++)
9      {
10         for (j = 2; j < i; j++)
11         {
12             if (i % j == 0)
13             {
14                 break;
15             }
16         }
17         if (j == i)
18         {
19             if (cnt % 5 == 0)
20             {
21                 printf("\n");
22             }
23             printf("\t%d", i);
24             cnt++;
25         }
26     }
27     return 0;
28 }
29

```

二、(25 分) 编写程序，由键盘读入一个 3 阶方阵 (3 行 3 列)，将每一行的最大值与该行位于主对角线上的元素进行互换，最后按行打印出变换后的方阵。要求：数组大小使用 #define 定义。[要求：输入、输出要有提示]

输入示例：

请输入 3 阶方阵 (3 行 3 列)：

```

1 2 3
4 5 6
9 8 7

```

输出示例：

变换后的矩阵为：

```

3 2 1
4 6 5
7 8 9

```

解：

参考代码：

```

1      #include<stdio.h>
2      void swap(int* a, int* b)
3      {
4          int* temp = *a;
5          *a = *b;

```

```

6      *b = temp;
7  }
8  int main()
9  {
10     #define n 3
11     int a[n][n];
12     int max;
13     int* m = NULL;
14     printf("请输入3阶方阵(3行3列): ");
15     for (int i = 0; i < n; i++)
16     {
17         for (int j = 0; j < n; j++) {
18             scanf("%d", &a[i][j]);
19         }
20     }
21     for (int i = 0; i < n; i++)
22     {
23         max = a[i][0];
24         m = &a[i][0];
25         for (int j = 0; j < n; j++) {
26             if (a[i][j] > max)
27             {
28                 m = &a[i][j];
29                 max = a[i][j];
30             }
31         }
32         int g = *m;
33         swap(m, &a[i][i]);
34     }
35     printf("变换后的矩阵为:\n");
36     for (int i = 0; i < n; i++)
37     {
38         for (int j = 0; j < n; j++) {
39             printf("%d\t", a[i][j]);
40         }
41         printf("\n");
42     }
43     return 0;
44 }

```

三、(30 分) 在主函数中，通过键盘读入字符串 (最大长度不超过 30, 包含字母、数字、空格和其他字符)，利用自定义函数 findlet 和 count 对字符串进行处理后，在主函数中输出：提取后的大写字母字符串，以及该字符串中大写字母出现的次数。其中：

(1) 函数 void findlet(char str[], char * let) 的功能: 提取字符串 str 中的所有英文字母，转换成大写，并依次存入字符串 let 中 (如字符串 str 是 “12a89Rry6u? k4X”，则函数完成后字符串 let 是 “ARRYUKX”)。

(2) 函数 void count(char *let) 的功能: 对字符串 let, 统计每一个大写字母出现的次数。

要求: 输入、输出要有提示

输入示例:

请输入长度不超过 30 的字符串:

12a89Rry6u? k4X

输出示例:

处理后的大写字母字符串为: ARRYUKX

字符 A 的出现次数为: 1

字符 R 的出现次数为: 2

字符 Y 的出现次数为: 1

字符 U 的出现次数为: 1

字符 K 的出现次数为: 1

字符 X 的出现次数为: 1

解:

参考代码:

```

1  #include<stdio.h>
2  #include <malloc.h>
3  int c[26] = { 0 };
4  void findlet(char str[], char* let)
5  {
6      int i, k;
7      int cnt = 0;
8      k = strlen(str);
9      for (i = 0; i < k; i++)
10     {
11         if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z'))
12         {
13             if (str[i] >= 'a' && str[i] <= 'z')
14             {
15                 str[i] = str[i] + ('A' - 'a');
16             }
17             let[cnt] = str[i];
18             cnt++;
19         }
20     }
21 }
22 void count(char* let)
23 {
24     int i, k;
25     k = strlen(let);
26     for (i = 0; i < k; i++)
27     {
28         c[let[i] - 'A'] += 1;
29     }
30 }

```

```

31  int main()
32  {
33      int i;
34      printf("请输入长度不超过30的字符串:\n");
35      char str[30];
36      char a[30] = { 0 };
37      int k;
38      gets(str);
39      k = strlen(str);
40      findlet(str, a);
41      k = strlen(a);
42      count(a);
43      printf("处理后的大写字符串为: ");
44      for (i = 0; i < k; i++)
45      {
46          printf("%c", a[i]);
47      }
48      printf("\n");
49      for (i = 0; i < 26; i++)
50      {
51          if (c[i] > 0) {
52              printf("字符%c的出现次数为:%d\n", i + 'A', c[i]);
53          }
54      }
55      free(c);
56      return 0;
57  }

```

四、(25 分) 程序功能: 时间转换。要求用结构体类型 struct time 表示 12 小时制的时间, 包含四个成员: 时、分、秒、am 或 pm。要求从程序中读入一个初始的 12 小时制的时间数 (例如: 11:59:59 am), 再读入一个秒数 n(n<60), 要求输出该时间再经过 n 秒后的时间值 (超过 12 点, 则 am 和 pm 发生转化)。[要求: 输入、输出要有提示]

输入示例:

请输入时间: 11:59:59 am

请输入延后的秒数 (<60): 59

输出示例:

再过 59 秒后的时间为: 0:0:58 pm

解:

参考代码:

```

1  #include <stdio.h>
2  #include <string.h>
3  int main(){
4      struct time {
5          int hour, minute, second;
6          char str[2];

```



```

7   }a;
8   int t;
9   printf("请输入时间: ");
10  scanf("%d:%d:%d%s", &a.hour, &a.minute, &a.second, &a.str);
11  printf("请输入延后的秒数(<60): ");
12  scanf("%d", &t);
13
14  a.second = a.second + t;
15  if (a.second >= 60) {
16      a.minute = a.second / 60 + a.minute;
17      a.second = a.second % 60;
18  }
19  if (a.minute >= 60) {
20      a.hour = a.minute / 60 + a.hour;
21      a.minute = a.minute % 60;
22  }
23  if (a.hour >= 12) {
24      a.hour = a.hour % 12;
25      if (strcmp(a.str, "am"))
26      {
27          strcpy(a.str, "am");
28      }
29      else
30      {
31          strcpy(a.str, "pm");
32      }
33  }
34  printf("再过%d秒后的时间为: %d:%d:%d%s", t, a.hour, a.minute, a.second, a.str);
35  return 0;
36  }

```