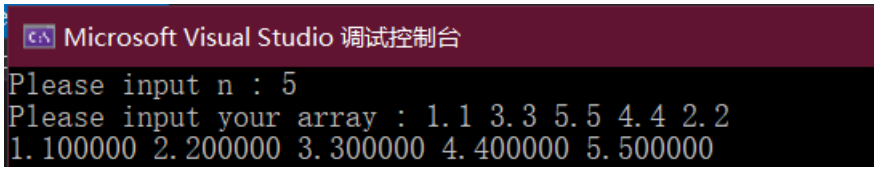
Exercise 1

- Write a program and a function to sort *n* numbers by the selection method, the type of numbers are double, and the numbers and the value of *n* are input from keyboard. It is requested that all the operations must use pointers.

```
#include <stdio.h>
#include <stdlib.h>
#include <float.h>

void selectSort(int* a, double* b);

void main() {
    int n;  //有n个数需要排序
    int* np = &n;
    double* nums = (double*)malloc(sizeof(double) * n);   //创建动态数组nums
    printf("Please input n : ");
    scanf("%d", np);
    printf("Please input your array : ");
    for (int i = 0; i < *np; i++) {
        scanf("%lf", nums + i);
    }
    selectSort(np, nums);   //进入选择排序函数
    for (int i = 0; i < *np; i++) {
        printf("%lf ", *(nums + i));
    }
    free(nums);
}

void selectSort(int* a, double* b)  //选择排序
{
    double min = DBL_MAX;
    int temp;
    for (int i = 0; i < *a; i++)
    {
        temp = i;
        for (int j = i; j < *a; j++)
        {
            if (*(b + j) < min)
            {
                min = *(b + j);
                temp = j;
```

```
            }
        }
        /*printf("%d\n", temp);*/
        *(b + temp) = *(b + i);
        /*printf("%f %f ", *(b + temp),min);*/
        *(b + i) = min;
        /*printf("%f \n", *(b + i));*/
        min = DBL_MAX;
    }
}
```

Case1:

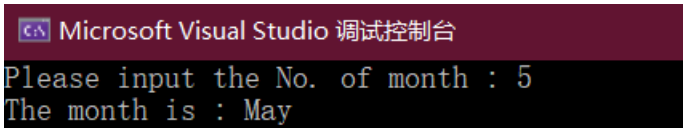


Exercise 2

- Write a program, input the number of month, then output the English name of that month. eg. If you input "3", then output "March". Use a pointer array to handle it.

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

void main() {
    char* monthName[12] =
{ "January","February","March","April","May","June","July","August","September","Octobe
r","November","December" };    //存入月数英文单词
    int i;   //月份数
    int* ip = &i;
    printf("Please input the No. of month : ");
    scanf("%d", ip);
    printf("The month is : ");
    printf("%s", *(monthName+*ip-1));  //输出对应单词
}
```

Case1:



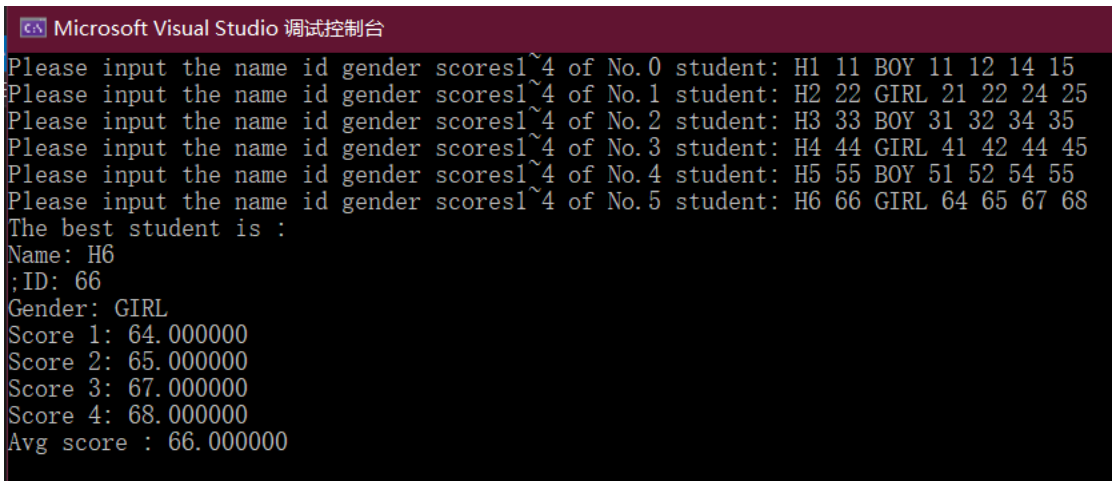
Exercise 3

- There are 6 students, each student's data including the student number, name, gender, and the scores of 4 courses. Initialize the 6 students' data, and output the average of the 4 courses for each student, and output the relevant information of the student with the maximum average scores (including the name, student number, gender, the scores of 4 courses, and the average score).

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

struct student    //结构体 存入学生数据
{
    char name[1000];
    char id[1000];
    char gender[1000];
    double scores[4];
    double avgScore;
}stu[6];

void main() {
    struct student* stup = stu;    //结构体指针
    int maxAvgi; //存放最大平均分坐标
    double maxAvg = INT_MIN;  //存放最大平均分
    for(int i=0;i<6;i++)
    {
        printf("Please input the name id gender scores1~4 of No.%d student: ",i);
        scanf("%s %s %s %lf %lf %lf %lf", (stup+i)->name, (stup + i)->id, (stup +
i)->gender, &(stup + i)->scores[0], &(stup + i)->scores[1], &(stup + i)->scores[2],
&(stup + i)->scores[3]);
        (stup + i)->avgScore = ((stup + i)->scores[0] + (stup + i)->scores[1] + (stup
+ i)->scores[2] + (stup + i)->scores[3])/4.0;
        if((stup+i)->avgScore>maxAvg)  //找出平均分最大的学生
        {
            maxAvg = (stup + i)->avgScore;
            maxAvgi = i;
        }
```

```
    }
    printf("The best student is : \n");
    printf("Name: %s\n;", (stup + maxAvgi)->name);
    printf("ID: %s\n", (stup + maxAvgi)->id);
    printf("Gender: %s\n", (stup + maxAvgi)->gender);
    printf("Score 1: %f\n", (stup + maxAvgi)->scores[0]);
    printf("Score 2: %f\n", (stup + maxAvgi)->scores[1]);
    printf("Score 3: %f\n", (stup + maxAvgi)->scores[2]);
    printf("Score 4: %f\n", (stup + maxAvgi)->scores[3]);
    printf("Avg score : %f\n", maxAvg);
}
```

Case1:



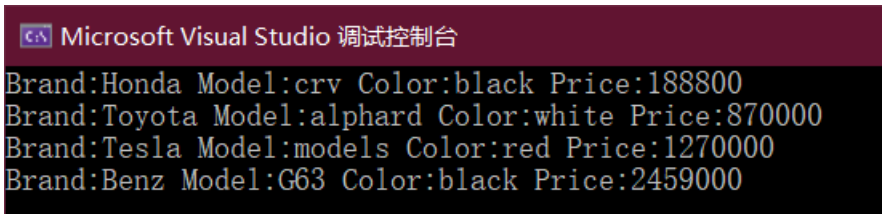
Exercise 4

- Write a program which defines a structure for car sale by the structure array, the member of the structure contains car name, model, color, and price. Initialize all the records in the main function, and write a function to sort the car records by price in ascending order, and output all the records of the cars.

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

struct carSell {    //结构体定义
    char name[20];
    char model[20];
    char color[20];
    int price;
```

```
};

int cmp(const void* a, const void* b) { //快速排序判断函数
    struct carSell xx = *(struct carSell*)a;
    struct carSell yy = *(struct carSell*)b;
    return xx.price - yy.price;
}

void main() {
    struct carSell car[4] =
{ {"Toyota","alphard","white",870000},{"Honda","crv","black",188800},{"Tesla","models",
"red",1270000},{"Benz","G63","black",2459000} }; //初始化
    struct carSell* carp = car; //结构体指针定义
    qsort(car, 4, sizeof(car[0]), cmp); //快速排序
    for (int i = 0; i < 4; i++)
    {
        printf("Brand:%s Model:%s Color:%s Price:%d \n", (carp+i)->name, (carp +
i)->model, (carp + i)->color, (carp + i)->price);  //结构体指针输出
    }
}
```

Case1:



Exercise 5

- Input a string by keyboard, this string contains numbers and nonnumeric characters. Regard the successive numbers as a whole, turn them into an integer, and store these integers into an integer array, count how many such numbers, and output these integer numbers one by one. Use pointers to complete the operations. For example:

  Input a string: njiop87h988h9sodfi osidjf7329?iju907&67

  (Then you should put the number 87 into a[0], put 988 into a[1], and put 9 into a[2], ……, and so on.)

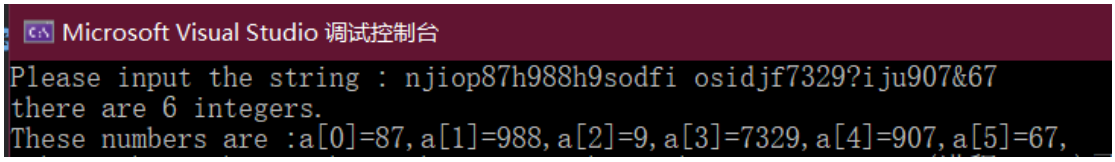
  The output result is: there are 5 integers.

  These numbers are: a[0]=87, a[1]=988, a[2]=9, a[3]=7329, a[4]=907, a[5]=67.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main() {
    char* str = (char*)malloc(sizeof(char) * 10000); //字符数组定义
    int* nums = (int*)malloc(sizeof(int) * 9999);    //储存整数
    int count = 0;    //计数器
    printf("Please input the string : ");
    scanf("%[^\n]", str);
    /*printf("%s\n", str);*/
    for(int i=0;i<strlen(str);i++) //逐字判断
    {
        if(str[i]<='9'&& str[i]>='0'){
            nums[count] = 0;
            /*printf("%c \n", str[i]);*/
            while (str[i] <= '9' && str[i] >= '0')
            {
                /*printf("%d ", nums[count]);*/
                nums[count] = (nums[count] * 10) + (str[i]-48);
                /*printf("%d \n", nums[count]);*/
                i++;
            }
            count++;
            i--;
        }
    }
```

```
    printf("there are %d integers. \n", count);
    printf("These numbers are :");
    for(int j=0;j<count;j++)
    {
        printf("a[%i]=%i,", j, nums[j]);
    }
    free(str);
    free(nums);
}
```

Case1:



Exercise 6

- Input num_str strings with different length (each string's length is uncertain and is less than 100) and store them into a two dimensional array, where num_str and the strings are input from the keyboard from the main function, you can use gets() or scanf() to input the string. Use a one dimensional pointer array to point these strings. Sort these strings in another function by strcmp() and the bubble method, and output these strings in the main function.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void mpSort(char str[][100],int n);

void main() {
    int num_str;  //数组大小
    printf("Please input the num_str : ");
    scanf("%d", &num_str);
    char (*str)[100] = (char(*)[100])malloc(sizeof(char) * num_str * 100); //动态定义字符串指针数组大小
    for(int i=0;i<num_str;i++)
    {
        printf("Please input the string %d/%d :", i + 1, num_str);
        scanf("%s", str[i]);
```

```
    }

    mpSort(str, num_str); //传入排序函数
    free(str);
}

void mpSort(char str[][100], int n)//冒泡排序
{
    char temp[100];
    for(int i=0;i<n-1;i++)//冒泡排序
    {
        for(int j=i;j<n-1;j++)
        {
            if (strcmp(str[j], str[j + 1])>0)
            {
                strcpy(temp, str[j]);
                strcpy(str[j], str[j + 1]);
                strcpy(str[j + 1], temp);
            }
        }
    }
    for (int i = 0; i <n; i++) //输出
    {
        printf("%s\n", str[i]);
    }
}
```

Case1: