1. Write a function to calculate the total pay of a waitress where the rate of pay (thousand VND / hour) and the number of hours. It is supposed that the number of hours is up 40, the waitress is paid at basic rate, and above 40 at double rate. Test your code in main function.
   a. The function:

```c
float salary(float basic, float hour){
    float total_pay, result;
    if (basic > 0){
        if ( 0 < hour && hour <= 40){
            total_pay = basic * hour;
        }else if (hour > 40){
            total_pay = basic * 40 + (hour - 40) * basic * 2;
        }else{
            total_pay = - 1;
        }
    }else{
        total_pay = -1;
    }
    result = total_pay;
    return result;
}

```

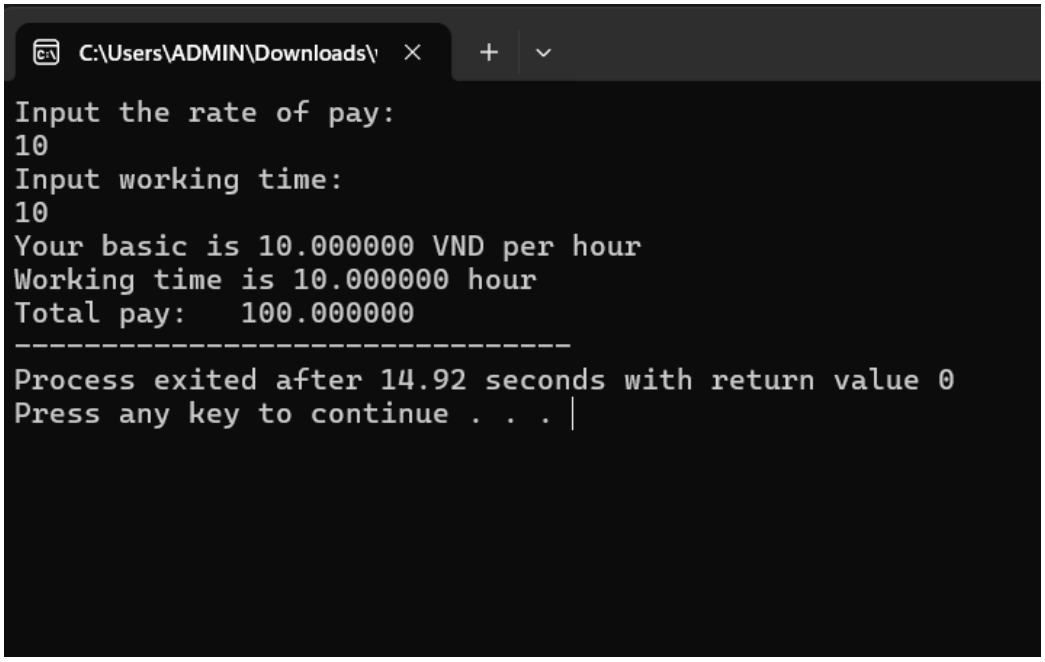
   b. Call the function in main function:

```c
float salary(float basic, float hour);
int main(){
    float bs, time, rs;
    printf("Input the rate of pay: \n");
    scanf("%f", &bs);
    printf("Input working time: \n");
    scanf("%f", &time);
    printf("Your basic is %f VND per hour\n", bs);
    printf("Working time is %f hour\n", time);

    rs = salary(bs,time);
    if (rs == -1){
        printf("error!!");
    }else{
        printf("Total pay:   %f", rs);
    }
    return 0;
}
```

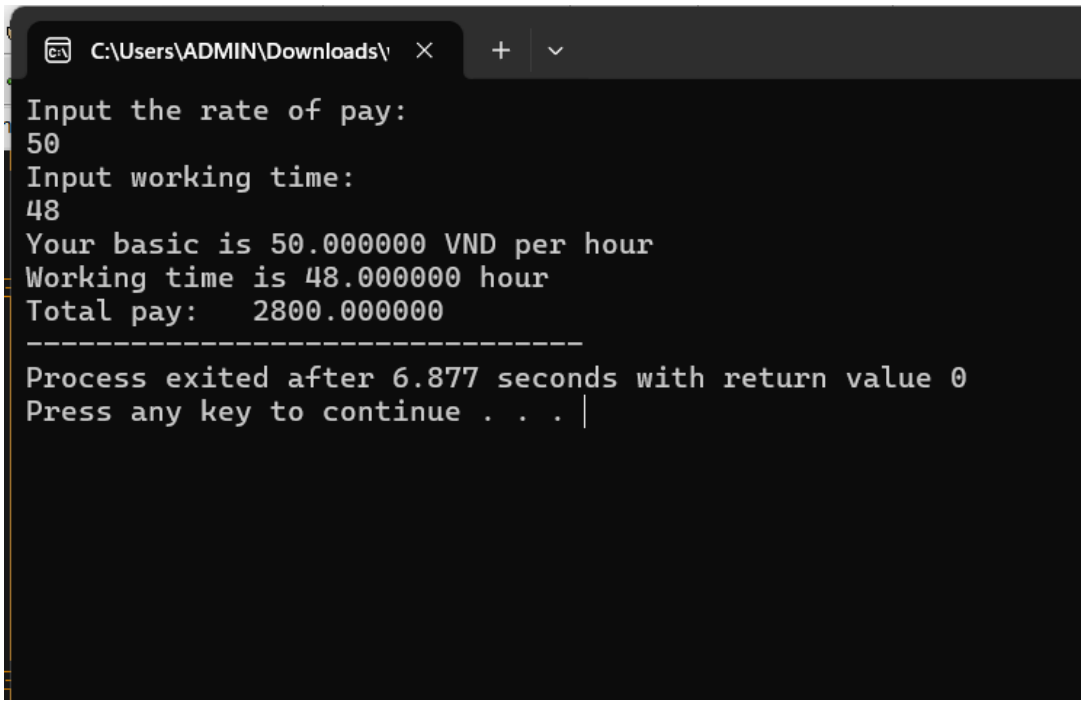
c. Test:

- Case 1:



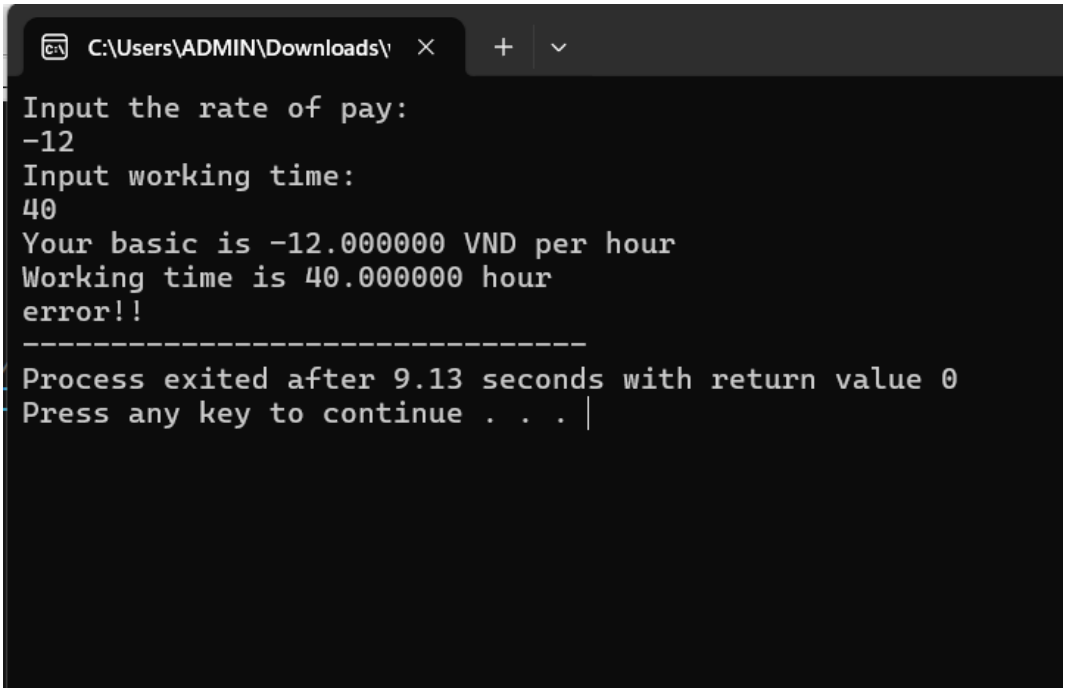
Walkthrough:
Line 16: enter 10 from keyboard →bs = 10
Line 18: enter 10 → time = 10
Line 22: pass bs & time to salary function →basic = 10, hour = 10
Line 33: hour = 10 →true→execute condition command
Line 44: assign total_pay to result variable
Line 45: function return result
Line 22: assign result of function to rs variable
Line 26: print out on screen

- Case 2:



Walkthrough:
Line 16: enter 50 → bs = 50
Line 18: enter 48 →time = 48
Line 31: pass bs and time variable to salary function
Line 36: true→ execute condition command
Line 45: the function return result

- Case 3:



Walkthrough:
Line 16: enter -12 → bs = -12
Line 18: enter 40 → time = 40
Line 31: pass bs and time variable to salary function
Line 33: false→execute line 41→assign -1 to total_pay
Line 45: function return result assigned by total_pay
Line 23: true → execute condition command
Line 24: print out on screen

2. Write a function to generate an array containing square numbers given the number of elements. Test your code in main function
   a. The function:

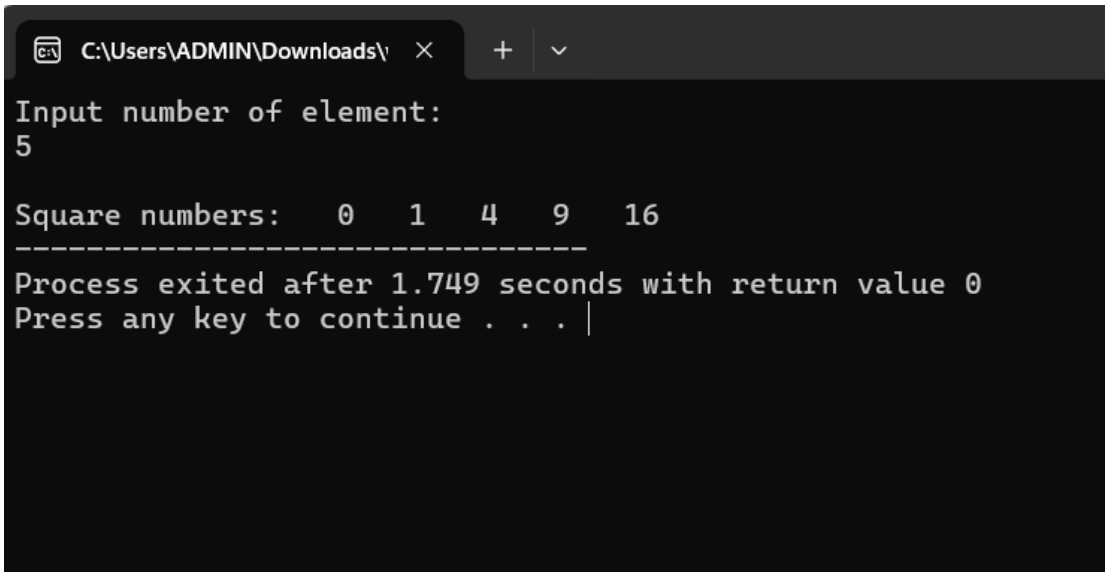
```
78  //build function
79  void sqr_num(int n, int result[]){
80      if (n>0){
81          for (int i = 0;i<=n-1;i++){
82              result[i] = i * i;
83          }
84      }else{
85          printf("please reinput!!");
86      }
87  }
88
```

b. Call the function in main function:

```c
63  void sqr_num(int n, int result[]);
64  int main(){
65      int a, rs[1000];
66      printf("Input number of element: \n");
67      scanf("%d", &a);
68      sqr_num(a,rs);
69      printf("\n");
70      if (a>0){
71          printf("Square numbers:   ");
72          for (int i=0;i<=a-1;i++){
73              printf("%d   ", rs[i]);
74          }
75      }
76      return 0;
77  }
```

c. Test:

- Case 1:



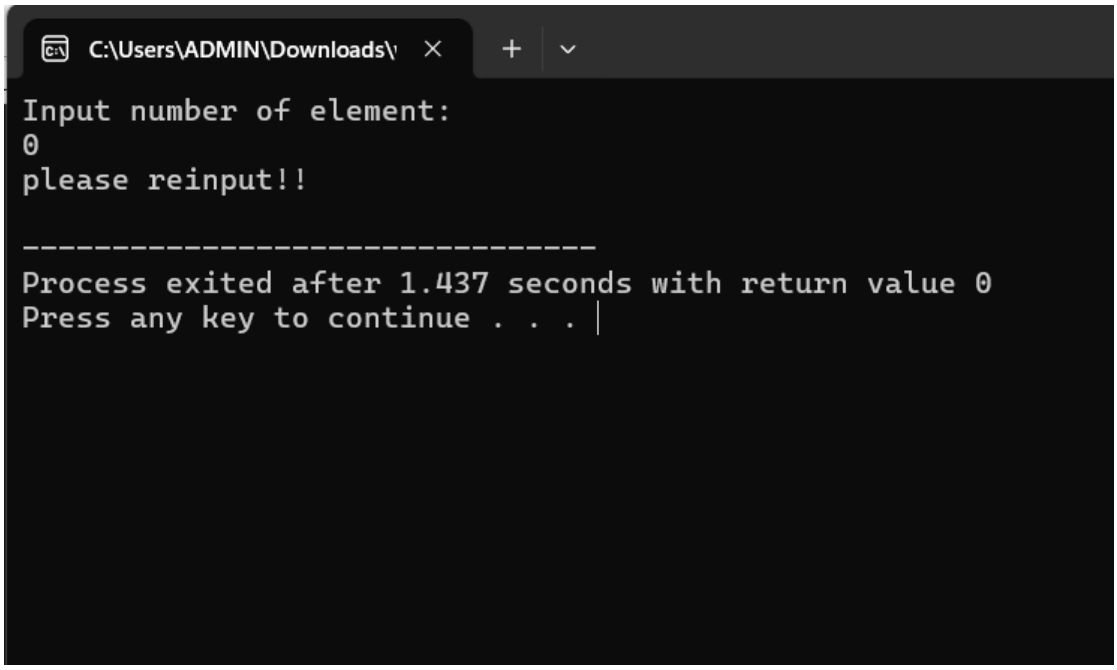
Walkthrough:
Line 67: enter 5 from keyboard →a = 5
Line 68: pass a to sqr_num function
Line 81: execute loop command →i=0→result[0] = 0*0 = 0
→i = 1 → result[1] = 1*1 = 1

→i = 2 → result[2] = 2*2 = 4

…

Line 73: print out on screen

- Case 2:



Walkthrough:

Line 67: enter 0 from keyboard →a = 0

Line 68: pass a to sqr_num function

Line 80: false → execute else command →line 85: print out on screen

3. Write a function to generate a new array containing the element of a given array where two elements k and k+1 in a given array are swapped. Test your code in main function.
   a. The function:

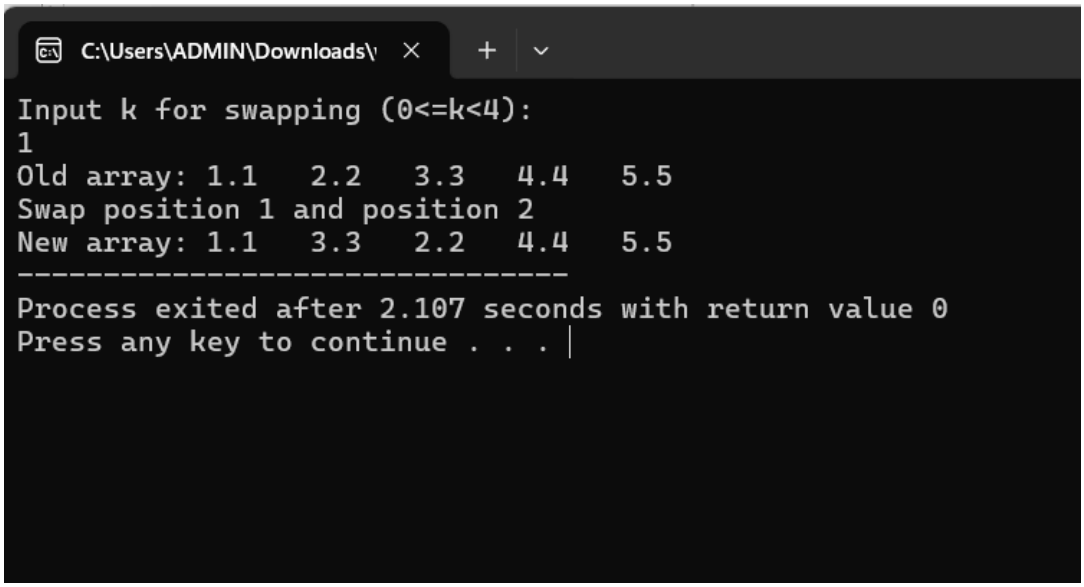
```
126  //build function
127  void swap_array(int k, float array[], float result[]){
128      float temp;
129      if (0 <= k && k < 4){
130          for (int i=0;i<5;i++){
131              if (i == k){
132                  temp = array[i];
133                  array[i] = array[i+1];
134                  array[i+1] = temp;
135              }
136              result[i] = array[i];
137          }
138      }else{
139          printf("please reinput!");
140      }
141  }
142
```

b. Call the function in main function:

```
105  void swap_array(int k, float array[], float result[]);
106  int main(){
107      float arr[5] = {1.1, 2.2, 3.3, 4.4, 5.5}, rs[5];
108      int a;
109      printf("Input k for swapping (0<=k<4): \n");
110      scanf("%d", &a);
111      swap_array(a, arr, rs);
112      if (0 <= a && a < 4){
113          printf("Old array: ");
114          for (int i=0;i<5;i++){
115              printf("%.1f   ", arr[i]);
116          }
117          printf("\n");
118          printf("Swap position %d and position %d\n", a, a+1);
119          printf("New array: ");
120          for (int i=0;i<5;i++){
121              printf("%.1f   ", rs[i]);
122          }
123      }
124      return 0;
125  }
```

c. Test:

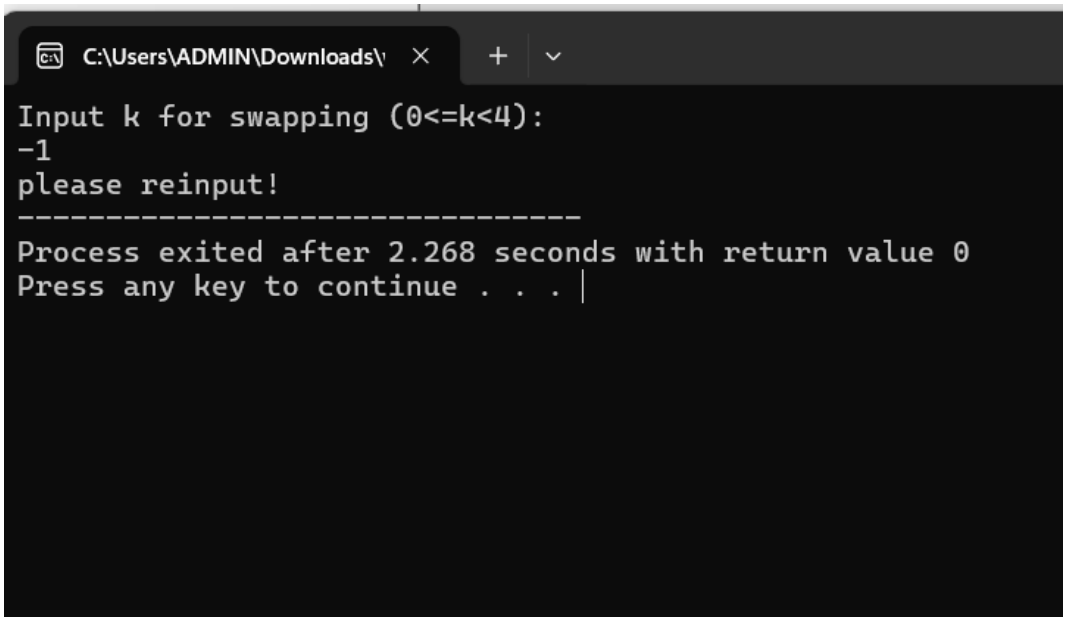
- Case 1:



Walkthrough:
Line 110: enter 1 → a =1
Line 111: pass a, arr, rs to swap_array function
Line 129: true → execute condition command → establish “temp” variable to swap
Line 136: store values of array into result

Line 120: print out on screen new array

- Case 2:



Walkthrough:

Line 110: enter -1 → a =-1

Line 111: pass a, arr, rs to swap_array function

Line 129: false → execute else command → print out on screen “please reinput”