Subject: PRF192

Workshop 03

Name: Hoàng Thủy Nguyên

Student ID: DE191056

Write a menu to call three following functions

1. *Function* 1: **Calculate personal income tax**

In Viet Nam, each people has to pay for yearly personal income tax. The general rule is if your income per month is less than or equal to 9 million VND, you will not pay. Otherwise you will pay. Specifically, if the income is from 9 000 001 VND to 15 000 000 VND, you must pay 10% of the amount of income that over 9 million VND. If the income is over 15 million VND, you must pay 20% of the amount of income that over 15 million VND. Write a program to calculate the tax that a person must pay, given that the her/his income is inputted from the keyboard.

a. The function is:

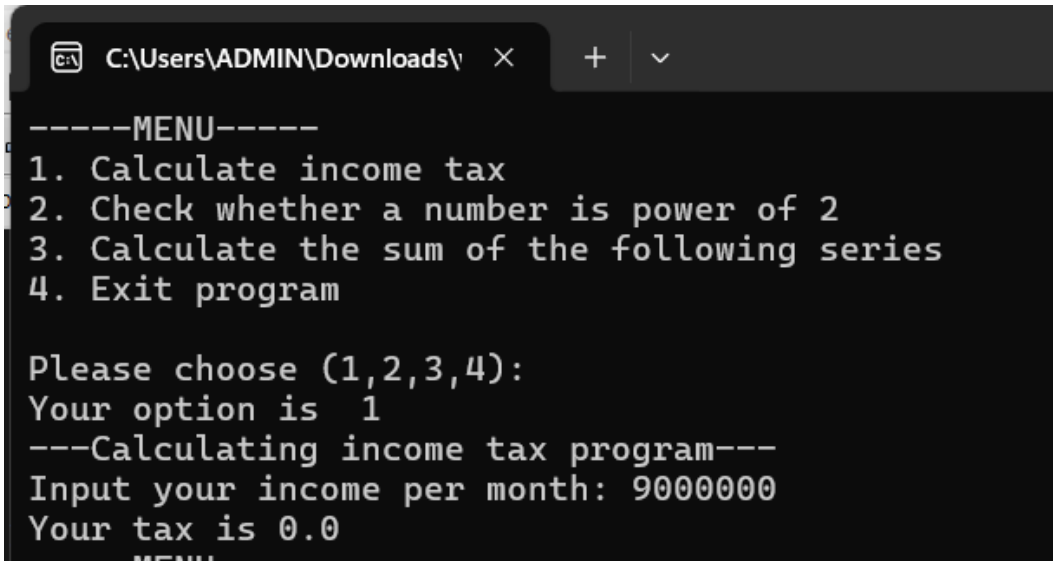
```
10 float cal_tax(int income){
11     float tax;
12     if (income > 0){
13         if (income <= 9000000){
14             tax = 0;
15         }else if(income > 9000000 && income <= 15000000){
16             tax = (income-9000000)*10.0/100.0;
17         }else{
18             tax = (income-15000000)*20.0/100.0;
19         }
20     }else{
21         tax = -1;
22     }
23     return tax;
24 }
25
```

b. Call the function in main function

```
94         switch (opt){
95             case 1:{
96                 //income tax
97                 int ic;
98                 float result;
99                 printf("---Calculating income tax program---\n");
100                printf("Input your income per month: "); scanf("%d", &ic);
101                result = cal_tax(ic);
102                if (result == -1){
103                    printf("error!\n");
104                }else{
105                    printf("Your tax is %f\n", result);
106                }
107                break;
108            }
```

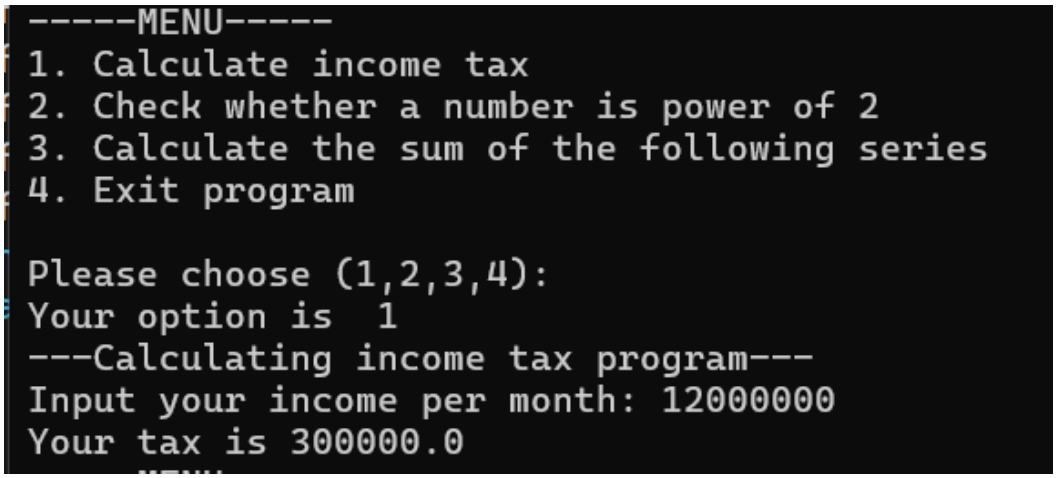
c. Test:

- Case 1:



Walkthrough:
Line 92: enter 1 → opt = 1 →execute case 1
Line 98: enter 9 000 000 → ic = 9 000 000
Line 10: pass value of ic into function cal_tax →"income" argument = 9 000 000
Line 12: true → execute condition command
Line 13: true → execute condition command → tax = 0
Line 23: function return tax
Line 99: assign result of cal_tax function into "result" variable
Line 102: because tax = 0 → false → execute else command→Line 103: print out on screen "Your tax is 0.0"
Line 144: because opt = 1(different from 4) →true→ execute loop command→ show menu

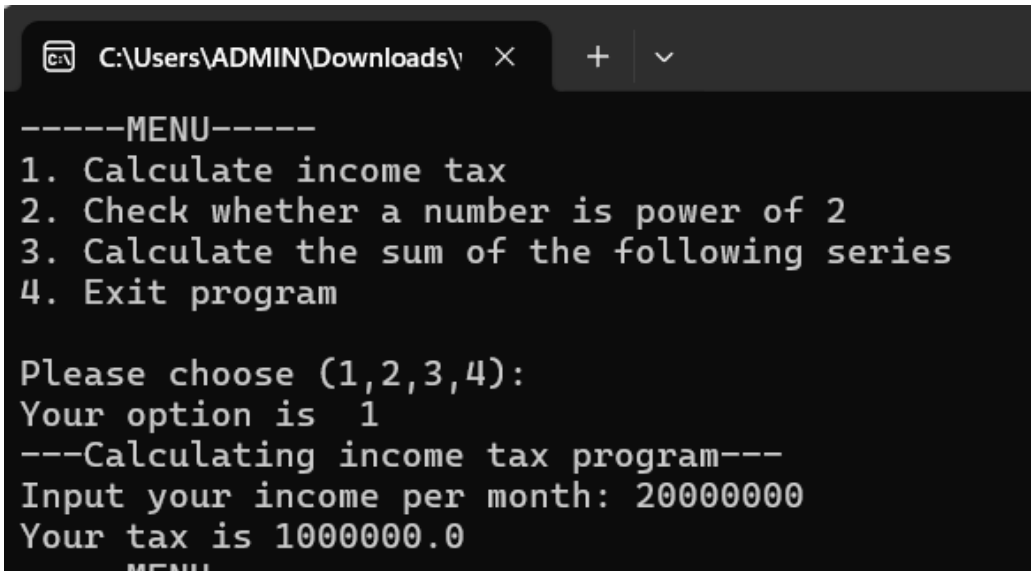
- Case 2:



Walkthrough:

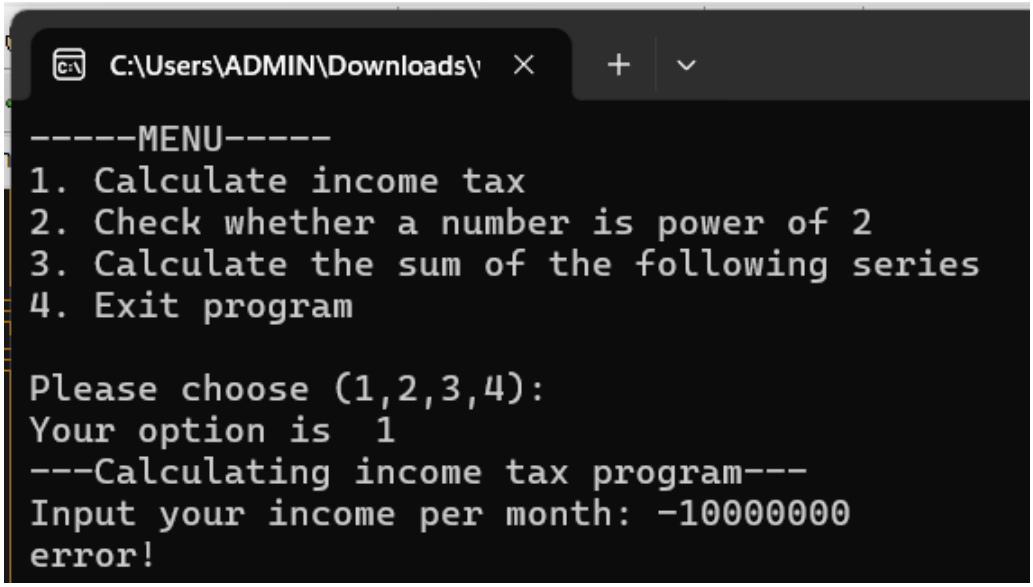
Line 92: enter 1 → opt = 1 →execute case 1
Line 98: enter 12 000 000 → ic = 12 000 000
Line 10: pass value of ic into function cal_tax →"income" argument = 12 000 000
Line 12: true → execute condition command
Line 15: true → execute condition command → tax = 300 000
Line 23: function return tax
Line 99: assign result of cal_tax function into "result" variable
Line 102: because tax = 300 000 → execute else command→Line 103: print out on screen "Your tax is 300 000"
Line 144: because opt = 1(different from 4) →true→ execute loop command→ show menu

- Case 3:



Walkthrough:
Line 92: enter 1 → opt = 1 →execute case 1
Line 98: enter 20 000 000 → ic = 20 000 000
Line 10: pass value of ic into function cal_tax →"income" argument = 20 000 000
Line 12: true → execute condition command
Line 18: true → execute condition command → tax = 1 000 000
Line 23: function return tax
Line 99: assign result of cal_tax function into "result" variable
Line 102: because tax = 1 000 000→ execute else command→Line 103: print out on screen "Your tax is 1 000 000"
Line 144: because opt = 1(different from 4) →true→ execute loop command→ show menu

- Case 4:



Walkthrough:
Line 92: enter 1 → opt = 1 →execute case 1
Line 98: enter -10 000 000 → ic = -10 000 000
Line 10: pass value of ic into function cal_tax →"income" argument = -10 000 000
Line 20: false → execute else command→ tax = -1
Line 23: function return tax
Line 99: assign result of cal_tax function into "result" variable
Line 100: because tax = -1→ execute condition command→Line 101: print out on screen "error!"
Line 144: because opt = 1(different from 4) →true→ execute loop command→ show menu

---

2. *Function* 2 : **Check whether a number is power of 2**

Input a number from the keyboard, then check whether the number is power of 2.

a. The function is:

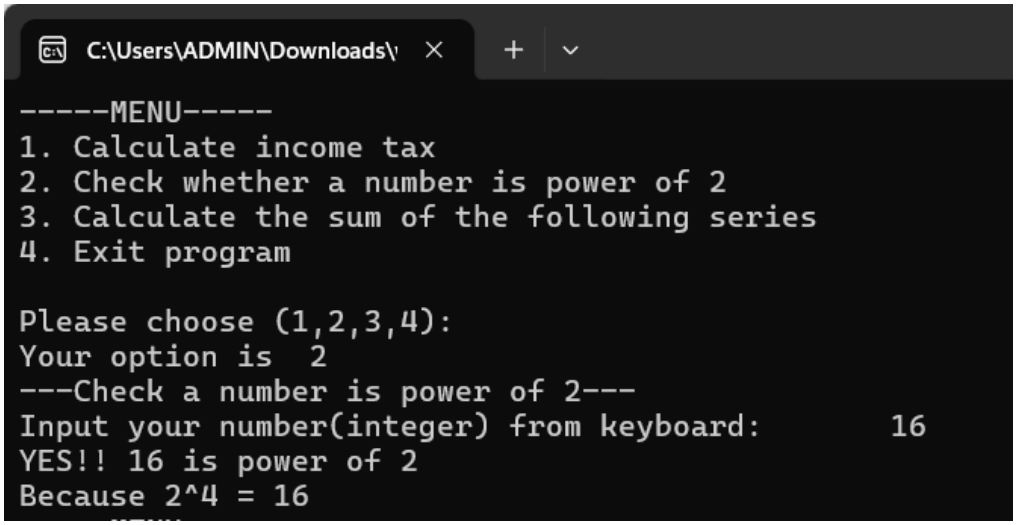
```c
//function 2 -->Check whether a number is po
/*Input a number from the keyboard, then che
void check_num(int a, int rs[]){
    int mu = 0;
    while (a > 0 && a%2==0){
        a /= 2;
        mu++;
    }
    if (a == 1){
        rs[0] = 1; //yes
    }else{
        rs[0] == 0; //no
    }
    rs[1] = mu;
}
```

b. Call the function in main function:

```c
        case 2:{
            //power of 2
            int num, res[2];
            printf("---Check a number is power of 2---\n");
            printf("Input your number(integer) from keyboard: \t"); scanf("%d", &num);
            check_num(num, res);
            if (res[0] == 1){
                printf("YES!! %d is power of 2\n", num);
                    printf("Because 2^%d = %d\n", res[1], num );
            }else{
                printf("NO!! %d is NOT power of 2\n", num);
            }
            break;
        }
```

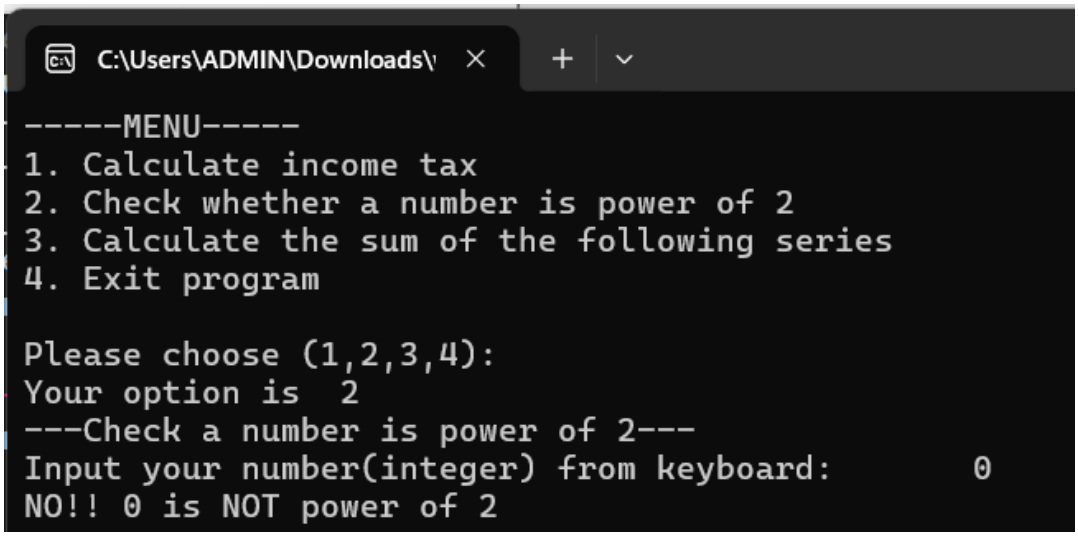
c. Test:

- Case 1:



Walkthrough:
Line 92: enter 2 → opt = 2 →execute case 2
Line 111: enter 16 → num = 16
Line 112: pass num=16 into check_num function→Line 30: a = 16
Line 32: true (because 16>0 and 16 is divisible by 2) →execute loop command
Line 33: a = 8(16/2), mu variable increases by 1
…(looping)
When a = 1 → end loop command
Line 36: true (a=1)→execute condition command→Line 37: assign 1 to rs[0] variable
Line 38: assign value of "mu" variable to rs[1]
Line 113: pass res[] into check_num function → res[0] = 1(because rs[0] = 1 of check_num function) →execute condition command→print out on screen "YES…"

- Case 2:



Walkthrough:
Line 92: enter 2 → opt = 2 →execute case 2
Line 111: enter 0 → num = 0

Line 112: pass num=0 into check_num function→Line 30: a = 0
Line 32: false → skip loop command
Line 39: false (a=0)→execute else command→Line 40: assign 0 to rs[0] variable
Line 113: pass res[] into check_num function → res[0] = 1(because rs[0] = 1 of check_num function) →execute condition command→print out on screen "NO…"

---

3. *Function* 3: **Calculate the sum of the following series.**
Given the sum as follows. Here, x and n are inputted from the keyboard.

a. The function is:

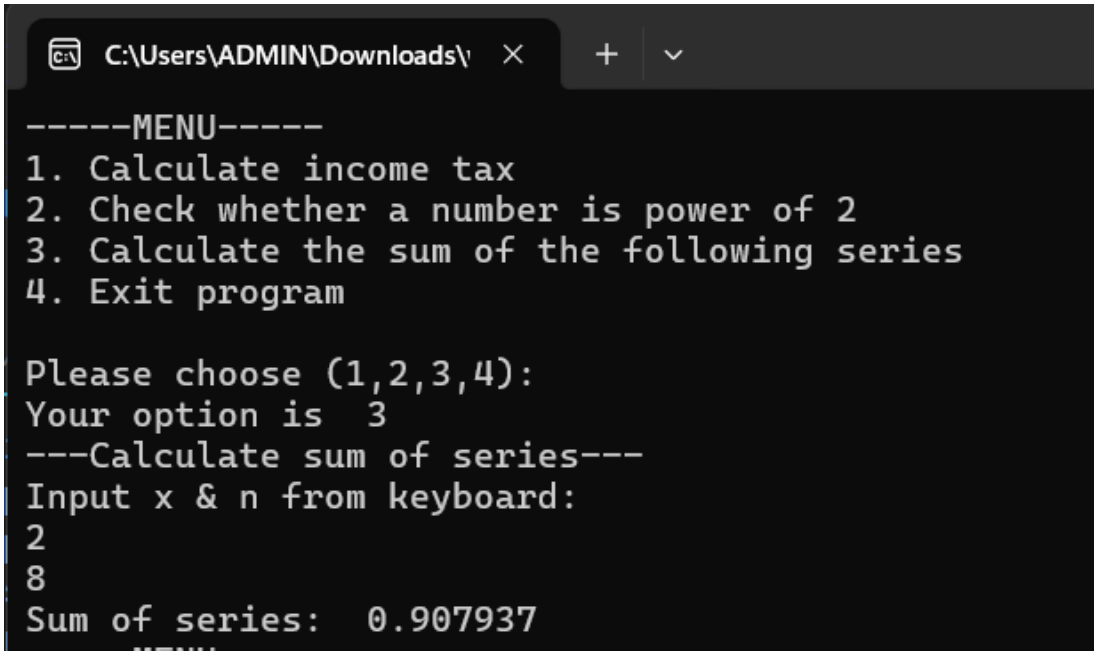
```
52 float sum_of(float x, int n) {
53     float sum = 0.0;
54     int gt;
55     if (n > 0) {
56         for (int i=1; i<=n; i=i+4) {
57             gt = 1;
58             for (int j=1; j<=i; j++) {
59                 gt *=j;
60             }
61             sum += pow(x,i)/gt;
62         }
63         for (int i=3; i<=n; i=i+4) {
64             gt = 1;
65             for (int j=1; j<=i; j++) {
66                 gt *=j;
67             }
68             sum -= pow(x,i)/gt;
69         }
70     } else {
71         sum = -1;
72     }
73     return sum;
74 }
```

b. Call the function in main function:

```
123             case 3: {
124                 //sum of series
125                 float result, a;
126                 int b;
127                 printf("---Calculate sum of series---\n");
128                 printf("Input x & n from keyboard: \t");
129                 scanf("%f%d", &a, &b);
130                 result = sum_of(a,b);
131                 if (result != -1) {
132                     printf("Sum of series:  %f\n", result);
133                 } else {
134                     printf("Error!!\n");
135                 }
136                 break;
137             }
```

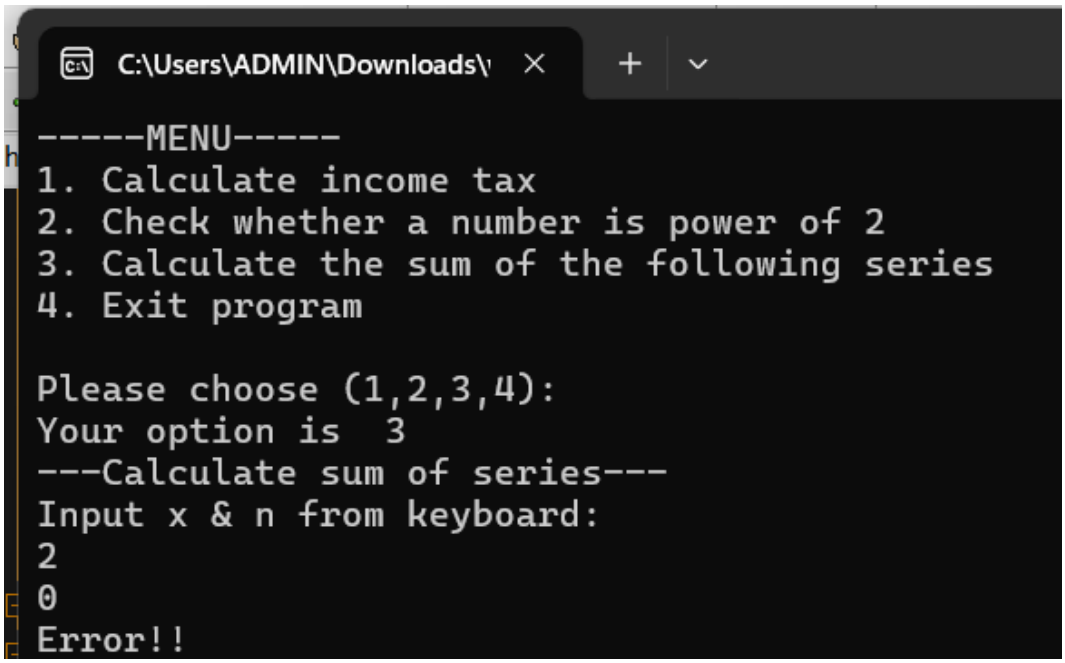
c. Test:

- Case 1:



Walkthrough:
Line 92: enter 3→opt = 3 → execute case 3:
Line 129: enter 2 8 → a=2, b=8
Line 130: pass a and b into sum_of function → Line 52: x = 2, n = 8
Line 55: true(8>0→execute condition command
Line 57→60: calculate factorial
Line 61: sum of positive elements in series
Line 63→69: calculate sum of series (including positive and negative elements of series)
Line 73: return sum(result after executing sum_of function)
Line 130: assign result of function to “result” variable
Line 132: print it out on screen ‘Sum of series is: 0.907937’

- Case 2:



Walkthrough:

Line 92: enter 3→opt = 3 → execute case 3:
Line 129: enter 2 → a=2, b=0
Line 130: pass a and b into sum_of function → Line 52: x = 2, n = 0
Line 70: false(0>0→execute else command
Line 71: assign -1 to sum variable
Line 73: return sum(which is result after executing sum_of function)
Line 130: assign result of function to “result” variable
Line 133: false(-1 = -1)→execute else command
Line 134: print it out on screen ‘error”