Workshop 1
Subject: PRF192
Name: Hoàng Thủy Nguyên
ID: DE191056

1. Write a function to check whether a number is prime. Then using this function to support printing out on screen the sequence of the primes from 2 to n where n is inputted from keyboard.
   - **Write a function to check whether a number is prime**

   a. The function is:

```c
//build function
int prime(int a){
    int result;
    if (a>=2){
        if (a==2){
            result = 1;
        }else if (a>2 && a % 2 != 0){
            result = 1;
        }else{
        result = -1;
        }
    }else{
        result = -1;
    }
    return result;
}

```

   b. Call the function in main function:

```c
#include <stdio.h>

/*1. Check prime number */
int prime(int a);
int main (){
    int n, rs;
    printf("Input any number from keyboard: \t");
    scanf("%d", &n);
    rs = prime(n);
    if (rs == 1){
        printf("PRIME!");
    }else {
        printf("NOT PRIME!");
    }
    return 0;
}
```

c. Test:

- The first case:

```
C:\Users\ADMIN\Downloads\

Input any number from keyboard:         33
PRIME!
--------------------------------
Process exited after 1.523 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:
Line 8: enter 33 → n = 33
Line 9: function “prime” is called →pass n to function “prime”
Line 18: a = 33
Line 24: result = 1 because 33 is prime
Line 9: result returned by “prime” function is assigned to “rs” variable
Line 11: because rs = 1 →print out PRIME

- The second case:

```
C:\Users\ADMIN\Downloads\

Input any number from keyboard:         0
NOT PRIME!
--------------------------------
Process exited after 1.709 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:
Line 8: enter 0 → n = 0
Line 9: function “prime” is called →pass n to function “prime”
Line 18: a = 0
Line 24: result = -1 because 0 is NOT prime
Line 9: result returned by “prime” function is assigned to “rs” variable
Line 11: because rs = -1 →print out NOT PRIME

- The third case:

```
C:\Users\ADMIN\Downloads\

Input any number from keyboard:         2
PRIME!
--------------------------------
Process exited after 0.7556 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:
Line 8: enter 2 → n = 2
Line 9: function "prime" is called →pass n to function "prime"
Line 18: a = 2
Line 24: result = 1 because 0 is prime
Line 9: result returned by "prime" function is assigned to "rs" variable
Line 11: because rs = 1 →print out PRIME

- **Then using this function to support printing out on screen the sequence of the primes from 2 to n where n is inputted from keyboard.**

a. The function is:

```
21 //build function
22 int prime(int a){
23     int result;
24     if (a>=2){
25         if (a==2){
26             result = 1;
27         }else if (a>2 && a % 2 != 0){
28             result = 1;
29         }else{
30         result = -1;
31         }
32     }else{
33         result = -1;
34     }
35     return result;
36 }
```

b. Call the function in main function:

```c
1  #include <stdio.h>
2
3  /*1. Check prime number */
4  int prime(int a);
5  int main (){
6      int n, rs;
7      printf("Input any number from keyboard: \t");
8      scanf("%d", &n);
9      if (n>1){
10         for (int i=2;i<=n;i++){
11             rs = prime(i);
12             if (rs == 1){
13                 printf("%d\n", i);
14             }
15         }
16     }else{
17         printf("There is no prime!");
18     }
19     return 0;
20 }
```

c. Test:

- The first case:

```
C:\Users\ADMIN\Downloads\

Input any number from keyboard:         10
2
3
5
7
9

--------------------------------
Process exited after 3.565 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:

Line 8: enter 10 → n=10

Line 9: because n=10>1 is true →execute condition command

Line 10: start with i=2(because 0 and 1 is not prime), pass i to “prime” function

Line 22: a = 2

Line 26: result = 1(because 2 is prime)

Line 11: result returned by “prime” function is assigned to “rs” variable

Line 13: print out 2 on screen because 2 is prime and increase i by one unit (i++)

Line 10: now i=3 and loop until i=n

- The second case:

```
C:\Users\ADMIN\Downloads\

Input any number from keyboard:         1
There is no prime!
--------------------------------
Process exited after 0.6085 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:
Line 8: enter 1 → n=1
Line 9: because n=1>1 is false →execute else command
Line 17: print out "There is no prime"

---

2. Write a function to calculate

$$\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+\cdots+\frac{1}{n!}$$

where n is inputted from keyboard.

a. The function is:

```
//build function
float sum_of_fac(int n){
    float sum;
    if (n>=0){
        sum = 1.0;
        for (int i=1;i<=n;i++){
            float gt = 1.0;
            for (int j=1;j<=i;j++){  //tinh giai thua
                gt *= j;
            }
            sum += 1.0/gt;
        }
    }else {
        sum = -1;
    }
    float result = sum;
    return result;
}

```

b. Call the function in main function:

```c
/*2. Calculate sequence n*/
float sum_of_fac(int n);
int main(){
    float rs;
    int a;
    printf("Input a positive integer from keyboard: ");
    scanf("%d", &a);
    rs = sum_of_fac(a);
    if (rs==-1){
        printf("Non-existent!");
    }else{
        printf("Sum of sequence is: %f", rs);
    }
    return 0;
}
```

c. Test:

- The first case:

```
C:\Users\ADMIN\Downloads\

Input a positive integer from keyboard: 3
Sum of sequence is: 2.666667
--------------------------------
Process exited after 1.384 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:

Line 44: enter 3 from keyboard → a=3

Line 54: pass a to n of "sum_of_fac" function→n=3

Line 56: because n=3>0 is true→ condition command is executed

Line 57: assign 1 to 'sum'

Line 58: initiate loop to calculate "sum"

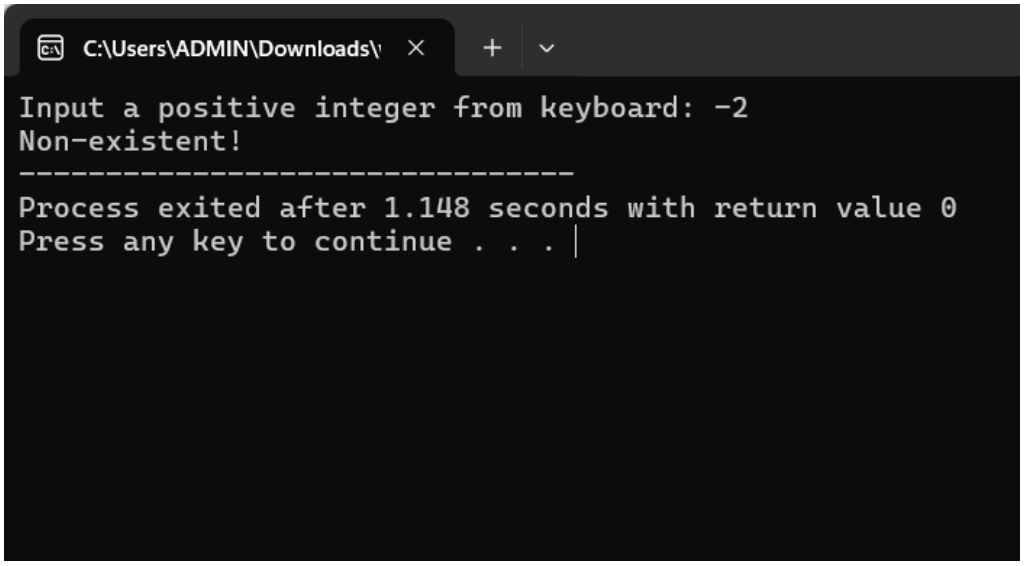
Line 60: initiate loop to calculate "gt"(factorial).

</>Explain loop in loop: First, i =1 → j=1 → gt = 1 = 1!

i=2 → j=1 → gt=1
j=2 → gt =1x2 = 2!
i=3 → j=1 → gt =1
j=2 → gt=1x2
j=3 → gt=1x2x3 =3!
…

Line 63: after calculate "gt"→ calculate "sum"
Line 69: : result returned by "sum_of_fac" function is assigned to "rs" variable
Line 49: print out "rs" on screen

- The second case:



Walkthrough:
Line 44: enter -2 from keyboard →a=-2
Line 54: pass a to n of "sum_of_fac" function→n=-2
Line 56: n=-2 >=0 is false →Line 65
Line 69: : result returned by "sum_of_fac" function is assigned to "rs" variable
Line 46: because "rs = -1"→ condition command is executed→print out screen "Non-existent"

3. Write a function to check whether a triangle is Equilateral.

Please test your program where walkthrough should be done along the lines of code.

a. The function is:

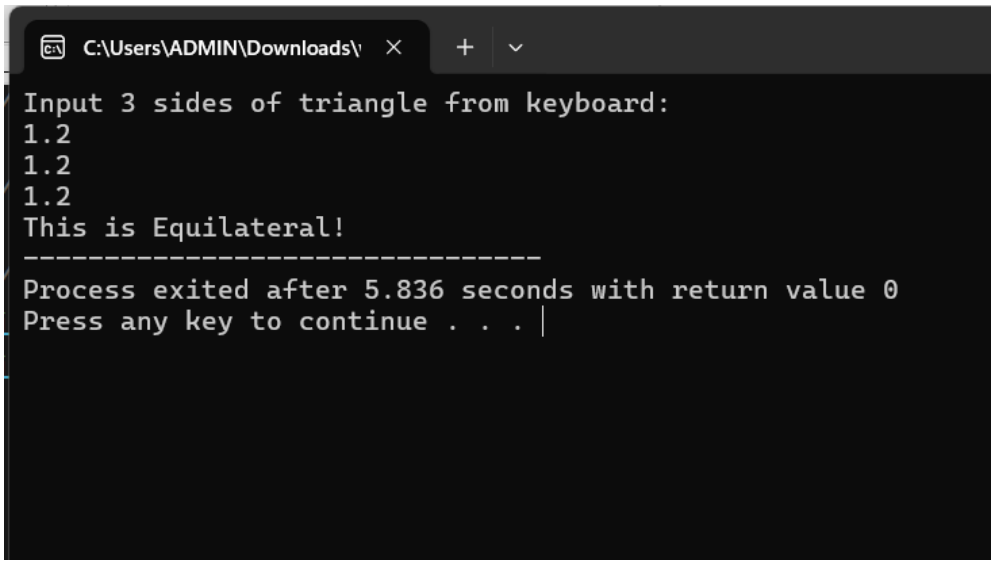
```c
//build function
int equi(float a, float b, float c){
    int result;
    if (a>0 && b>0 && c>0){ //dk cần
        if ( a+b>c && a+c>b && b+c>a){   //dk đủ
            if (a == b && a == c){
                result = 1;
            }else{
                result = -1;
            }
        }else{
            result = -2;
        }
    }else{
        result = -2;
    }
    return result;
}

```

b. Call the function in main function:

```c
/*3. Check triangle Equilateral*/
int equi(float a, float b, float c);
int main(){
    float a1, a2, a3;
    int rs;
    printf("Input 3 sides of triangle from keyboard: \n");
    scanf("%f%f%f", &a1, &a2, &a3);
    rs = equi(a1,a2,a3);
    if (rs == 1){
        printf("This is Equilateral!");
    }else if(rs==-1){
        printf("This is not Equilateral!");
    }
    else{
        printf("This is not triangle!");
    }
    return 0;
}
```

c. Test:

- The firtst case:



Walkthrough:
Line 80: enter 1.2 1.2 1.2 from keyboard → a1=1.2 a2=1.2 a3=1.2
Line 93: pass a1, a2, a3 to a, b, c of "equi" function→a=1.2 b=1.2 c=1.2
Line 95: a, b, c =1.2>0 is true→condition command is executed
Line 96: this condition is true →condition command is executed
Line 97: this condition is true→ condition command is executed →assign 1 to "result" variable
Line 108: result returned by "equi" function is assigned to "rs" variable
Line 82: because rs=1→ print out on screen "This is Equilateral"

- The second case:

```
C:\Users\ADMIN\Downloads\
Input 3 sides of triangle from keyboard:
1
3
4
This is not triangle!
--------------------------------
Process exited after 2.209 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:

Line 80: enter 1 3 4 from keyboard → a1=1 a2=3 a3=4

Line 93: pass a1, a2, a3 to a, b, c of "equi" function→a=1 b=3 c=4

Line 95: a, b, c >0(1>0, 3>0, 4>0) is true→condition command is executed

Line 96: this condition is false (1+3>4 is false) →line 102

Line 102: assign -2 to "result" variable

Line 108: result returned by "equi" function is assigned to "rs" variable

Line 84: because rs= -2 → print out on screen "This is not triangle"

- The third case:

```
C:\Users\ADMIN\Downloads\
Input 3 sides of triangle from keyboard:
-4
-5
-7
This is not triangle!
--------------------------------
Process exited after 6.541 seconds with return value 0
Press any key to continue . . .
```

Walkthrough:

Line 80: enter -4 -5 -7 from keyboard → a1=-4 a2=-5 a3=-7

Line 93: pass a1, a2, a3 to a, b, c of "equi" function→a=-4 b=-5 c=-7

Line 95: this condition is false (-4>0 is false)→Line 105

Line 105: assign -2 to “result” variable
Line 108: result returned by “equi” function is assigned to “rs” variable
Line 87: because rs=-2→ print out on screen “This is not triangle”