**Subject: PRF192- PFC**

**Workshop 04**

**Objectives:**

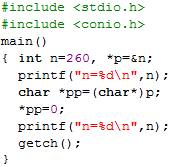
1. Managing data using pointers
2. Developing programs using simple menus

**Part 1: Use notebook**

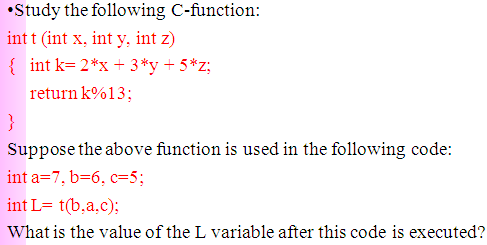
**Exercise 1** (1 mark) : Explain outputs:

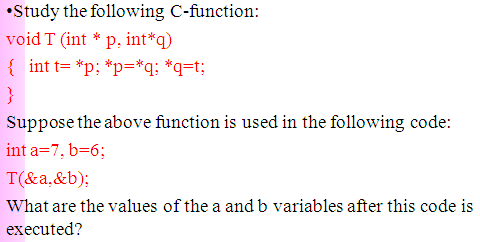


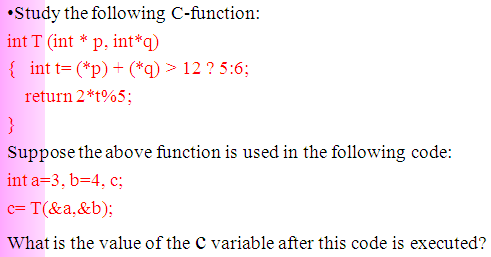
**Exercise 2: (1 marks) What are outputs**



**Exercise 3: (2 marks) Walkthroughs**







**Part 2: Develop a program using simple menu**

**Program 1(3 marks):**

|  |  |
| --- | --- |
| **Objectives** | Practice implementing a program with simple menu. |
| **Related knowledge** | None |
| **Problem** | Write a C program that will execute repetitively using a simple menu as following:   1. **Process primes** 2. **Print min, max digit in an integer;** 3. **Quit**   **Select an operation:**   1. When user selects the option 1, the program will accept a positive integral number and print out a message about whether the input number is a prime or not. 2. When user selects the option 2, the program will accept a positive integral number and print out the minimum and maximum digit in this number. 3. The program will terminate when user selects the option 3. |
| **Analysis** | **Nouns:**  - positive integral number 🡪 **int n**  - A number represents a choice of user 🡪 **int choice;**  **Functions**:  **int prime( int n) 🡪 see above**  **void printMinMaxDigits( int n) 🡪 see above** |
| **Suggested algorithm (logical order of verbs)** | Begin  Do /\* Print out the menu and get user choice\*/  { Print out “1- Process primes\n”;  Print out “2- Print min, max digit in an integer \n”;  Print out “3- Quit\n”;  Print out “Select an operation:”;  switch(choice)  { case 1: do  { Input n;  }  while(n<0);  If ( prime(n)==1) Print “ It is a prime\n”;  Else Print “ It is not a prime\n”;  break;  case 2: do  { Input n;  }  while(n<0);  printMinMaxDigits( int n) ;  break;  }  }  while ( choice >0 & choice<3);  End |

**Program 2(3 marks): ( refer to the workshop 2 for algorithms)**

Write a C program that will execute repetitively using a simple menu as following:

**1-Fibonacci sequence**

**2-Check a date**

**3-Quit**

**Choose an operation:**

1- When the option 1 is selected, the program will accept a positive integral number, called as n, then the first n Fibonacci numbers will be printed out

2- When the option 2 is selected, the program will accept a date then the program will tell that whether this data is valid or not.

3- If the option 3 is selected, the program quits

**More Programs**

You can pick 2 or 3 functions in the workshop 2, associate them to a new program.

**Exercise 1 :**

Ví dụ 1 :

Ta có n = 7 , m = 6

Con trỏ n = địa chỉ n = 7

Con trỏ m = địa chỉ m = 6

Phép tính : con trỏ n = con trỏ m + 2\*địa chỉ m – 3 \* địa chỉ n = 6 + 2\*6 – 3\*7 = -3

Con trỏ m = con trỏ m – con trỏ n ( đã trỏ tới địa chỉ n mới ) = 6 –(-3) = 9

m + n = -3 + 9 = 6

Ví dụ 2 :

Ta có c1 = A , c2= F

Con trỏ 1 = địa chỉ c1 = A

Con trỏ 2 = địa chỉ c2 = F

Phép tính : con trỏ 1 = con trỏ 1 + 3 = A+3 = c1 ( mới )

Con trỏ 2 = con trỏ 2 – 5 = F – 5 = c2 (mới )

c1 – c2 = A+3 – (F – 5) = A+3 -F+5 = A-F + 8

Trong bảng mã ASCII , A = 41 , F = 46 => A – F + 8 = -5 + 8 = 3

Ví dụ 3 :

Ta có x = 3.2 , y = 5.1

Con trỏ 1 = địa chỉ x = 3.2

Con trỏ 2 = địa chỉ y = 5.1

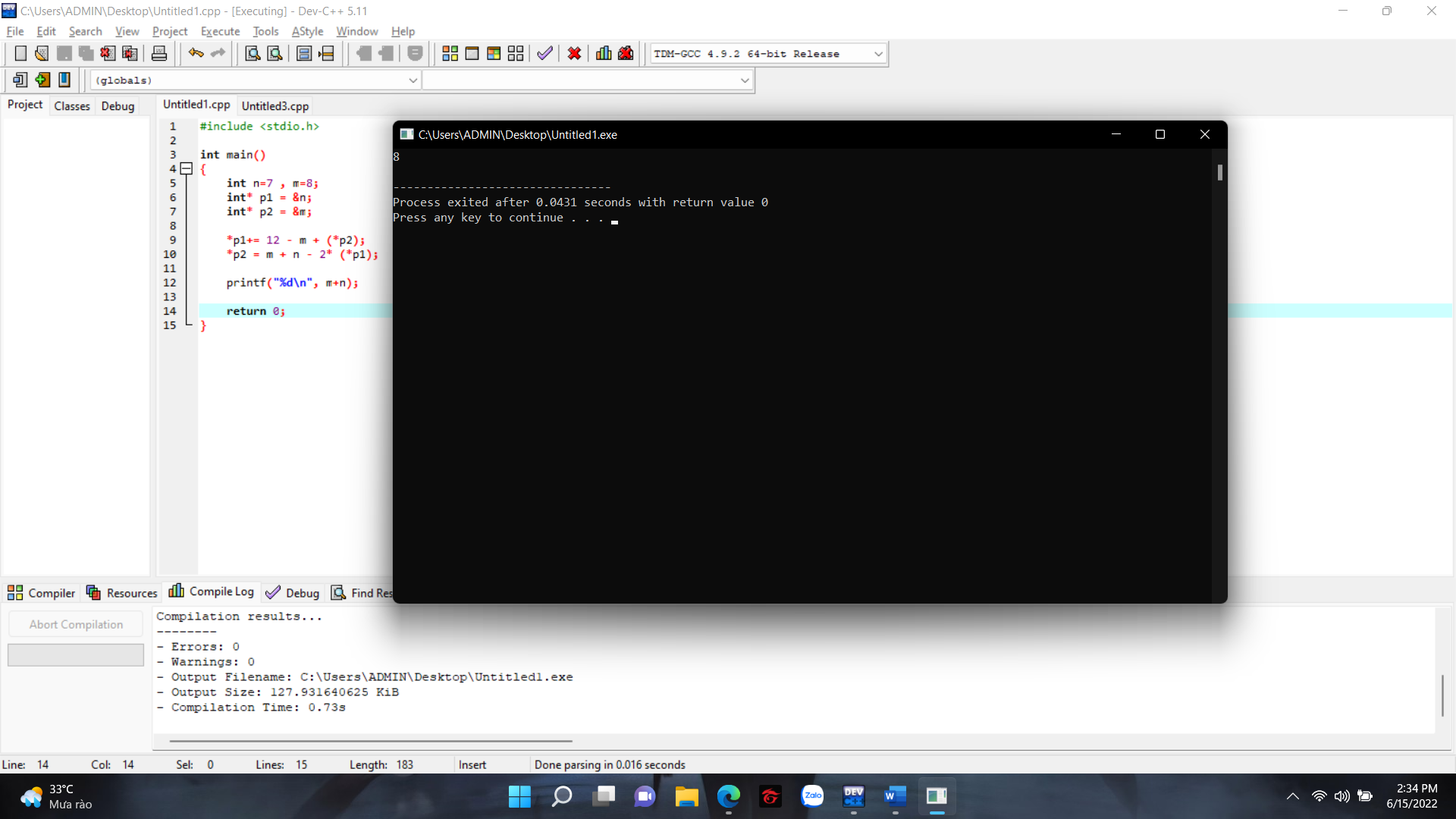
Phép tính : con trỏ 1 = con trỏ 1 + 3 – 2\*con trỏ 2 = 3.2 + 3 – 2\*5.1 = -4 = x ( mới )

Con trỏ 2 = con trỏ 2 – 3 \*(con trỏ 1 ) = 5.1 – 3\*(-4) = 17.1 = y ( mới )

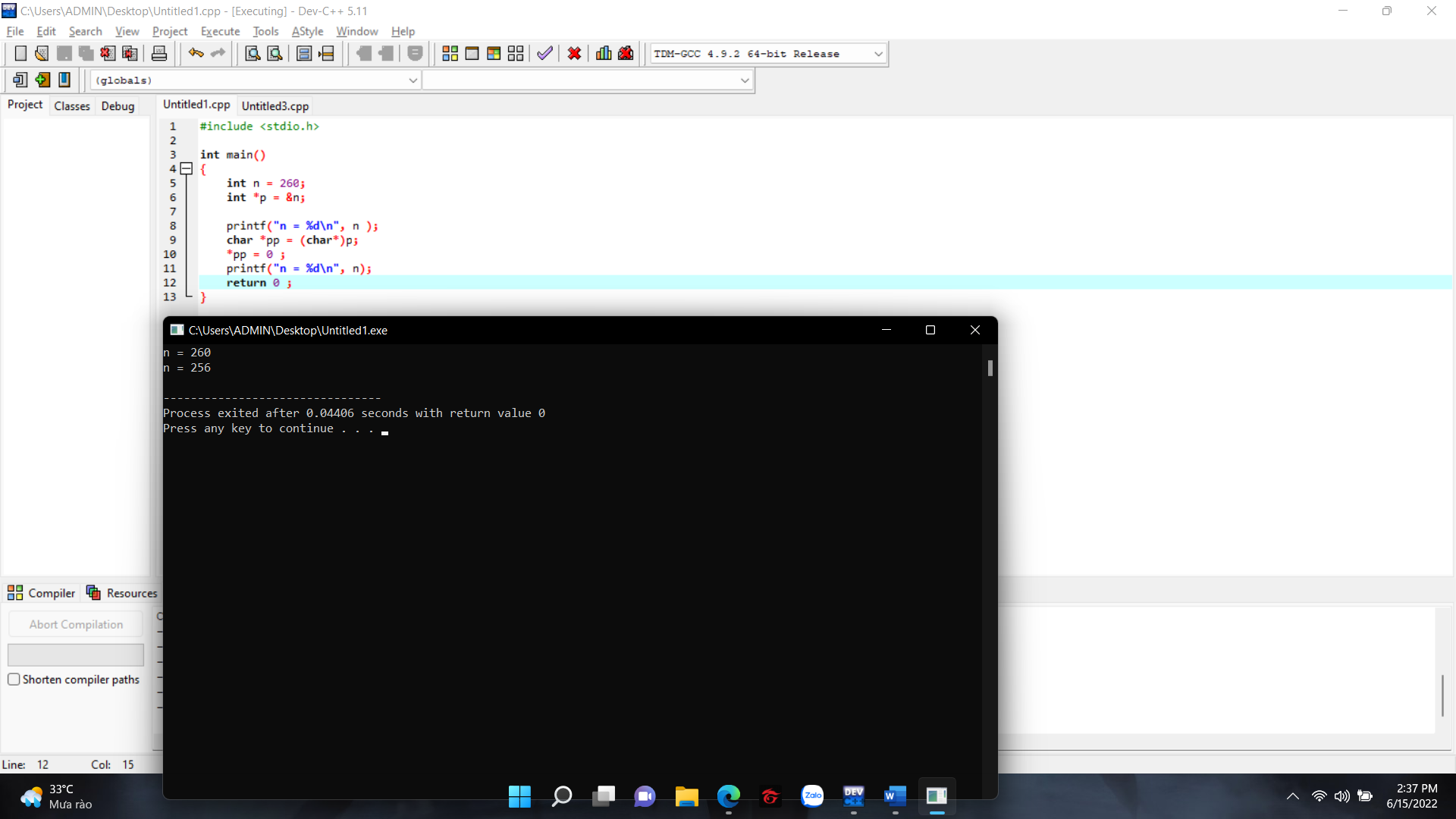
x + y = -4 + 17.1 = 13.1

**Exercise 2 :**

Output = 8



Output n = 260 , n = 256



**Program 1 :**

#include <stdio.h>

#include <stdlib.h>

void printMinMaxDigits (int n)

{

int digit , min , max ;

digit = n % 10 ;

n = n / 10 ;

min = max = digit ;

while (n >0){

digit = n % 10 ;

n = n / 10 ;

if (min > digit)

min = digit ;

if (max < digit)

max = digit ;

}

printf("Max digit : %d\n",max);

printf("Min digit : %d\n",min);

}

int prime (int n)

{

int i ;

if (n<=1)

return 1;

for (i=2 ; i \* i <= n ; i++)

{

if (n % i == 0)

return 1 ;

}

return 0;

}

int main()

{

int n = 0 , choice;

do

{

printf("1 - Process prime\n2 - Print min , max digit in an integer\n3 - Quit\n");

printf("Select an operation\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

do

{

printf("Please input integer :");

scanf("%d", &n);

}

while (n<0);

if (prime(n) == 1 )

printf("It is not a prime\n");

else

printf("It is a prime\n");

break;

case 2:

do

{

printf("Please input integer : ");

scanf("%d", &n);

}

while (n < 0);

printMinMaxDigits(n);

break;

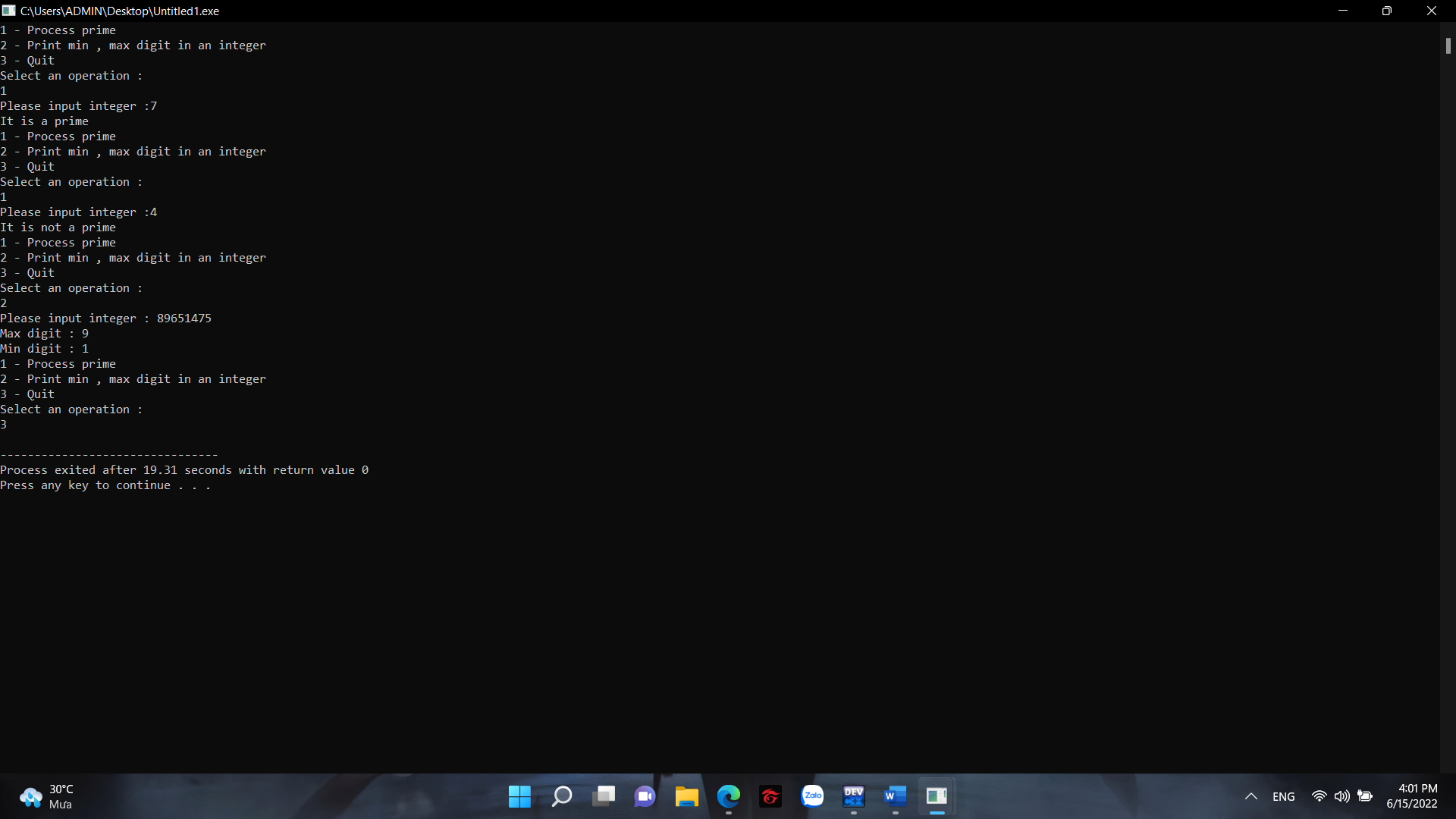
}

}

while ( choice > 0 & choice < 3);

return 0;

}



**Program 2 :**

#include <stdio.h>

#include <stdlib.h>

int fibo (int c)

{

int t1 = 1 , t2 = 1 , f = 1 , i ;

if (c == 1)

return t1;

for (i = 3; i<= c ;i++)

{

f = t1 + t2;

t1 = t2;

t2 = f;

}

return f;

}

int validDate (int d , int m , int y)

{

int maxD = 31;

if ((d >= 1 && d <= 31) && (m < 1 && m > 12))

{

return 0;

}

if (m == 4 || m == 6 || m == 9 || m == 11)

{

maxD = 30;

}

else if (m ==2 )

{

if ( y % 100 == 0 || (y % 4 == 0 && y % 100 !=0))

{

maxD = 29;

}

else

{

maxD = 28;

}

}

return d <= maxD;

}

int main()

{

int n = 0 , choice;

int d , m , y;

int c;

do

{

printf("\n1 - Fibonacci sequence\n2 - Check a date\n3 - Quit\n");

printf("Choose an operation : \n");

scanf("%d", &choice);

switch(choice)

{

case 2:

printf("Please input day/month/year :");

scanf("%d / %d / %d", &d ,&m ,&y );

if (validDate(d , m , y))

{

printf("Valid\n");

}

else

{

printf("Invalid\n");

}

break;

case 1:

printf("Input n position : ");

scanf("%d", &c);

while (c < 1)

{

printf("Your input is smaller than 1 , please input n again : ");

scanf("%d", &c);

}

printf("%d\n", fibo(c));

break;

default:

return 0;

}

}

while (choice > 0 & choice < 3);

return 0;

}

