**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:



Output = 6 vì \*pn trỏ vào n mà \*pn = \*pm + 2\*m - 3\*n = 6+2\*6-3\*7 = -3 => n = -3

\*pm trỏ vào m mà \*pm -= \*pn = 6 – (-3) = 9 => m = 9

Output m+n = 9 + (-3) = 6



C1 = A = 10

C2 = F = 15

\*p1 trỏ đến c1 mà \*p1 += 3 = 10+3 = 13 => c1 =13

\*p2 trỏ đến c2 mà \*p2 -= 5 = 15-5 =10 => c2 =10

Output c1 – c2 = 13 -10 = 3



Output = 19.5 vì \*p1 trỏ vào x mà \*p1 = 3-2\*(\*p2) = 3-2\*5.1 = -7.2=> x = -7.2

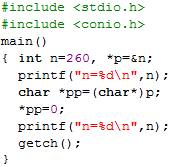
\*p2 trỏ vào y mà \*p2 -= 3\*(\*p1) = 5.1-3\*(-7.2) = 26.7 => y = 26.7

Output x+y = -7.2 + 26.7 = 19.5

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

Output = 8

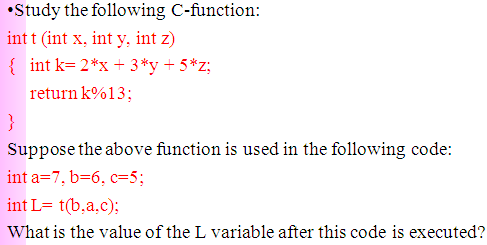
Output = 8



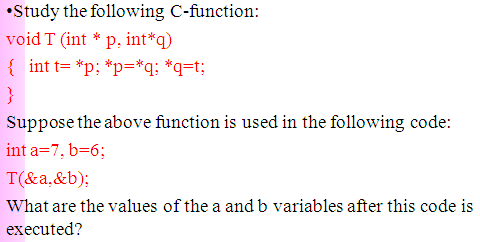
n=260

n=256

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

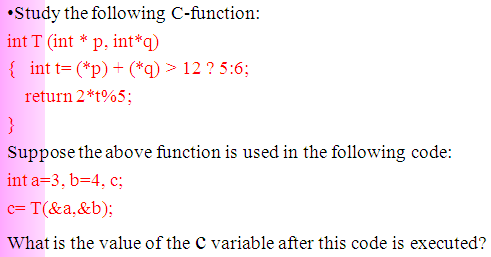


L = 12



A = 6

B = 7



C = 2 vì 3+4=7>12 là sai => t = 6 return 2\*6%5 = 12/5 = 2 => c = 2

**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 |

#include <stdio.h>

int snt(int n){

if (n <= 1 ){

return 0;

}

int i;

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

if (n % i == 0) {

return 0;

}

}

return 1;

}

void chuso(int n){

int min = 9;

int max = 0;

while (n > 0) {

int a;

a = n % 10;

if (a > max) max = a;

if (a < min) min = a;

n = n / 10;

}

printf("Chu so nho nhat la: %d\n", min);

printf("Chu so lon nhat la: %d\n", max);

}

int main() {

int c, n;

printf("1. Kiem tra so nguyen to\n");

printf("2. Tim chu so nho nhat, lon nhat\n");

printf("3. Quit\n");

printf("Nhap lua chon cua ban: ");

scanf("%d", &c);

switch (c) {

case 1:

do {

printf("Nhap so: ");

scanf("%d",&n);

} while (n < 0);

if (snt(n) == 1)

printf("So nay la so nguyen to");

else

printf("So nay la so nguyen to");

break;

case 2:

do {

printf("Nhap so: ");

scanf("%d",&n);

} while (n < 0);

chuso(n);

break;

default:

printf("Quiting");

break;

}

while (c != 3){

return 0;

}

}

**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.

#include <stdio.h>

void fibonacci(int n) {

int so1 = 0, so2 = 0, son;

printf("day fibonacci: ");

int i;

for (i = 0; i < n; i++) {

if (i <= 1)

so1 = i;

else {

son = so1 + so2;

so1 = so2;

so2 = son;

}

printf("%d ", son);

}

printf("\n");

}

int ktngay(int ngay, int thang, int nam) {

if (nam < 0)

return 0;

if (thang < 1 || thang > 12)

return 0;

if (ngay < 1)

return 0;

if (thang == 2) {

if ((nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0)) {

if (ngay > 29)

return 0;

} else {

if (ngay > 28)

return 0;

}

} else if (thang == 4 || thang == 6 || thang == 9 || thang == 11) {

if (ngay > 30)

return 0;

} else {

if (ngay > 31)

return 0;

}

return 1;

}

int main() {

int c;

int n;

int ngay, thang, nam;

do {

printf("1. Day fibonacci\n");

printf("2. Kiem tra ngay thang nam\n");

printf("3. Quit\n");

printf("Nhap lua chon: ");

scanf("%d", &c);

switch (c) {

case 1:

printf("Nhap so nguyen duong: ");

scanf("%d", &n);

if (n <= 0)

printf("So khong hop le\n");

else

fibonacci(n);

break;

case 2:

printf("Nhap ngay thang nam: ");

scanf("%d %d %d", &ngay, &thang, &nam);

if (ktngay(ngay, thang, nam))

printf("Ngay thang nam hop le\n", ngay, thang, nam);

else

printf("Khong hop le\n");

break;

default:

break;

}

}

while (c != 3);

return 0;

}