Subject: PRF192-
Workshop 1: Input/Output, computations and basic logics Subject

PART 1:

Program 1:

Write a C program that prints the following text on the screen:

Hello World!

Program 2: Write a C program to print your name, date of birth. and Address.

Program 3: VIết 1 chương trình hiển thị tổng, hiệu, tích, thương của 2 số 446 và 223 trên màn hình.

Yêu cầu: mỗi biểu thức hiển thị trên một dòng.

Output:

446 + 223 = 669

446 - 223 = 223

446 * 223 = 99458

446 / 223 = 2

Program 4: Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.

Expected Output:

```
######
#
#
#####
#
#
#
```

Program 5: Write a program to create 2 integer variables a and b, assign value 125 to a and 600to b then display the following result as output.

Output:
a+b =725

Program 6: Write a program to declare 2 variable a and b which are real numbers, assign value 10.5 to a, 7.3 to b and display the following line as output,

Output:
a/b=1.44

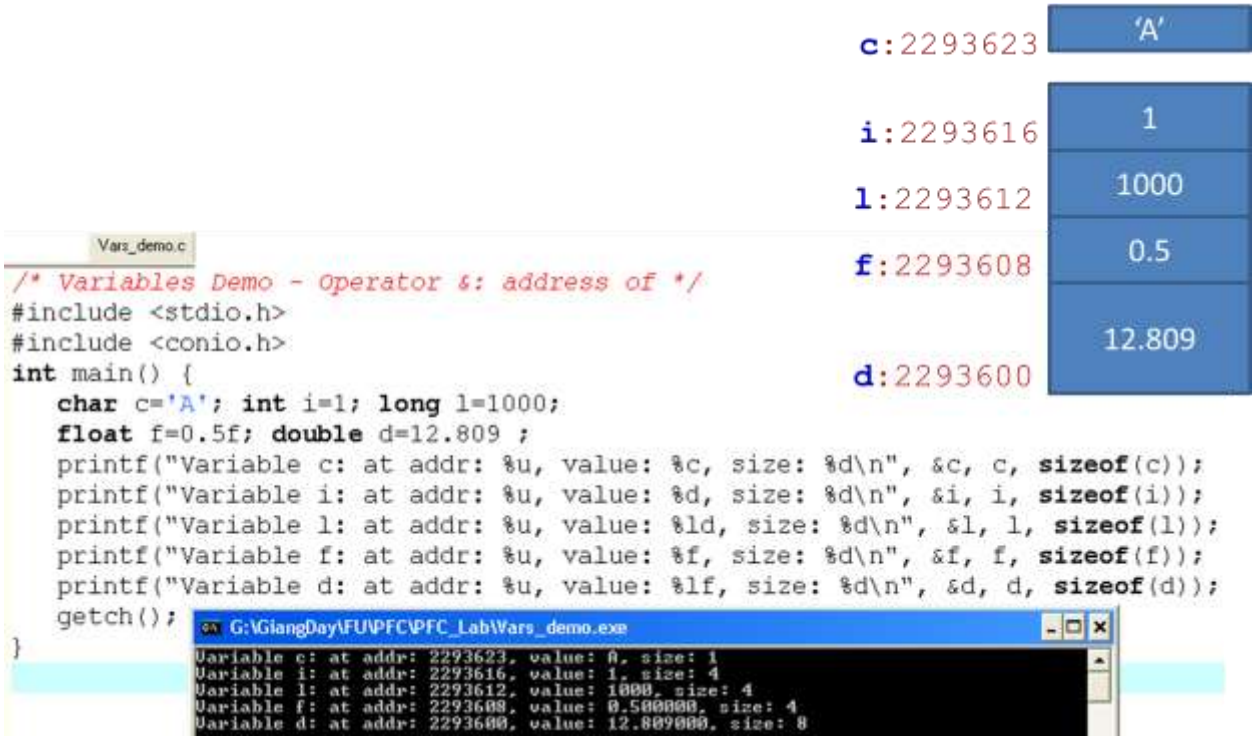
Program 7: Give a rectangle with the length of 7.8 and the width of 3.6. Write a program to print the area of this rectangle on the screen as below:

Area= 28.0800

Program 8: Write a program to display character 'd' on the screen.

```c
#include<stdio.h>
int main()
{
    char d='a' + ...;
    printf("%c",d);
    return 0;
    }
```

Program 9:
Sample




```c
#include <stdio.h>
int n;
double x;
char c1;
int main()
{   int m;
    short s;
    long L;
    float y;
    printf("Code of main:%u\n", &main));
    printf("Variable n, add:%u, memory size:%d\n", &n, sizeof(n));
    /* Your code to view address and memory size of other variables*/
    /* Complete the program, compile and run it */
    /* Draw the memory of the program*/
    getchar();
    return 0;
}
```

Program 10:

: Enter two numbers. Then, the sum of these two integers is calculated and displayed on the screen.

Program 11:

: Enter two numbers. Then print swap them.

Program 12:
: Swap Numbers Without Using Temporary Variables **(homework)**

CONDITIONAL LOGIC

Program 13:
: Write a program to accepts an integer n from the user than check whether n is an even or odd number.

Program 14:
:Write a program that accepts an integer n from the user then checks the following conditions:

If n is a positive integer, print " n is a positive integer"

If n is a negative integer, print" n is a negative integer"

If n is equal to 0, print " n is equal to 0".

Program 15:
: Write a program to read two integers a and b then check whether both a and b are not equal to zero or not.

If two values are not equal to 0, print the following line on the screen:

```
a is not equal to 0 and b is not equal to 0
```

If a or b is equal to 0, print the following line on the screen:

```
a is equal to 0 or b is equal to 0
```

Program 16:

: Write a program that accepts three integers from the user and prints the biggest number among them on the screen.

For example, if you enter three numbers as below:

```
4 7 3
```

When the code is compiled and executed, it produces the following result:

```
7
```

If you enter:

```
4 4 4
```

When the code is compiled and executed, it produces the following result:

```
4
```

Program 17:

: Write a program to read an integer a then check whether a is in the range [10, 100] or not.

If a is in the range [10, 100], print the following line on the screen:

```
{P} is in range (10, 100)
```

If a is not in the range, print the following line on the screen: :

```
{P} is not in range (10, 100)
```

where {P} is the value of a.

Program 18:

: Write a program that accepts the test score of a student (knowing that the valid score is greater than or equal to 0 and less than or equal to 10) and checks whether the entered score is valid or not.

If the score is valid, print the following line on the screen:

```
The score is valid
```

If the score is invalid, print the following line on the screen:

```
The score is not valid
```

Program 19: Given 2 integer variables `a` and `b` and a character variable `c` knowing that `c` is one of 4 characters `'+', '-', '*', '/'`. Write a program to read 3 variables `a, b` and `c` then display the result of expression when applying the operation `c` on `a` and `b`.

For example, if `a = 7, c = '+', b = 9`, enter the following line:

```
7 + 9
```

When the code is compiled and executed, it produces the following result:

```
16
```

LOOP: FOR

Program 20: Write a program that accepts an integer `n` from the user then displays all the numbers from `1` to `n` on the screen.

For example, if you enter `10` from the keyboard, the program will produce the following result:

```
1 2 3 4 5 6 7 8 9 10
```

***Program* 21**: Write a program that accepts two integers `a` and `b` from the user and displays all numbers from `a` to `b` on the screen.

For example, if `a = 5, b = 9`, the screen will display as below:

```
5 6 7 8 9
```

***Program* 22:** Write a program that accepts an integer `n` from the user then displays all numbers from `n` to `-n (n >= -5)` in descending order.

For example, if `n = 5`, the screen will display as below:

```
5 4 3 2 1 0 -1 -2 -3 -4 -5
```

***Program* 23**: Write a program that accepts two integers `a` and `b` from the user and displays the sum of all the numbers from `a` to `b` on the screen:

For example, if `a = 5, b = 9`, the screen will display as below:

```
35
```

Because `5 + 6 + 7 + 8 + 9 = 35`

***Program* 24**: Write a program that accepts an integer `n` from the user and displays the sum of all odd numbers from `0` to `n` on the screen.

For example, if `n = 7`, the program will produce the following result:

```
16
```

Because `1 + 3 + 5 + 7 = 16`

***Program* 25**: Write a program that accepts two integers `a` and `b` from the user and prints all the numbers from `a` to `b`, which are divisible by 3:

For example, if `a = 1, b = 20`, the program will display on the screen as below:

```
3 6 9 12 15 18
```

***Program* 26**: Write a program that accepts an integer `n` from the user and displays the result of `n!` on the screen.

For example, if `n = 5`, the program will display on the screen as below:

```
120
```

Because `1 * 2 * 3 * 4 * 5 = 120`.

Program **27**: Write a program that accepts an integer `n` from the user and prints the divisors of `n (n > 0)` on the screen.

For example, if `n = 12`, the screen will display as below:

```
1 2 3 4 6 12
```

LOOP: WHILE AND DO-WHILE

Program **28**:**:** Write a program that accepts an integer `n` and prints all even numbers from `n` to `100` on the screen.

For example, if `n = 90`, the program will produce the following result:

```
90 92 94 96 98 100
```

Program **29**:**:** Write a progarm that accepts an integer `n` from the user then prints all divisors of `n` on the screen.

For example, if `n = 12` , the program produces the following result:

```
6
```

Because all divisors of `12` are `1, 2, 3, 4, 6, 12`

Program **30**:**:** Write a program that accepts two integers `a` and `b` from the user then prints the result of a^b on the screen.

For example, if `a = 2, b = 3`, the program will produce the following result:

```
8
```

Because `2 * 2 * 2 = 8.`

Program **31**:**:** Write a program that accepts two integers `a` and `b` from the user then prints all numbers from `a` to `b`, which are divisible by `3` and `5`.

For example, if `a = 1, b = 50`, the program produces the following result:

```
15 30 45
```

***Program* 32::** Fill in the blank (`...`) to complete the program that prints all numbers from `1` to `50` on the screen.

```c
#include<stdio.h>

int main() {
    for (int i = 1; i <= 100; i++) {
        ...
        printf("%d ", i);
    }
    return 0;
}
```

***Program* 33::** Fill in the blank (`...`) to complete the program that prints all odd numbers from `1` to `100`.

```c
#include<stdio.h>

int main() {
    for (int i = 1; i <= 100; i++) {
        ...
        printf("%d ", i);
    }
    return 0;
}
```

***Program* 34::** Write a program that prints numbers from `1` to `5` using `do-while` loop.

```c
#include<stdio.h>

int main() {
    int i = 1;
    do {
        ...
    } while (...);
    return 0;
}

```

Program **35**:: Write a program to print all numbers from `1` to `1000` (including `1` and `1000`), which end with `0`. It means the program will display `10, 20, 30, ..., 990, 1000` on the screen.

```c
#include<stdio.h>

int main() {
    int i = 1;
    do {
        ...
    } while (i <= 1000);
    return 0;
}
```

PART 2:

Program 1 (2 marks)

Write a program that allows user inputting a simple expression containing one of four operators +, -, *, / then the result is printed out to the monitor. Input format: num1 operator num2,
An example of user interface
Enter an expression (+ - * /): 4*5
Result: 20

Sample Analysis

	Content	Implementation
Nouns	Expression, format num1 operator num2	double num1, num2 char op double result

	result	
Verbs	Begin Accept num1, op, num2 **Calculate result** Print out result End	scanf("%lf %c%lf", &num1, &op, &num2) switch (op) { case '+' : result = num1 + num2; print out result; break; case '-' : result = num1 - num2; print out result; break; case '*' : result = num1 * num2; print out result; break; case '/' : if (num2==0) print out "Divide by 0 " else { result = num1 / num2; print out result; } break; default: print out "Op is not supported" }

Implement this program.

Program 2 (2 marks) – Yearly Personal Income Tax

Suppose that:
In Viet Nam, each people has to pay for his/her yearly personal income tax as the following description:

Rules:

Tax-free income:
Personal pending amount (tiền nuôi bản thân) **pa=** 9 000 000$/month
Alimony (tiền cấp dưỡng) for each his/her dependent **pd=** 3 600 000$/month/dependent
With **n** dependents, Yearly tax-free income: **tf = 12*(pa + n*pd)**

Taxable income (thu nhập chịu thuế)
ti = income – tf
(If ti<=0 then income tax = 0)

Based on taxable income, the employee has to pay his/her income tax with levels pre-defined in the following table:

Level	*Taxable Income*	*Income tax*
1	Less than or equal to 5.000.000	5%
2	From 5.000.001 to 10.000.000	10%
3	From 10.000.001 to 18.000.000	15%
4	Over 18.000.000	20%

```
if(ti<=5tr)
        It=ti*0.05;
else if(ti<=10tr)
        it=5tr*0.05+(ti-5tr)*0.1
else if(ti<=18tr)
        it=5tr*0.05+ 5tr*0.1+(ti-10tr)*0.15
else
        it=5tr*0.05+ 5tr*0.1+8tr*0.15+(ti-18tr)*0.2
```

```
Ti= 3tr                 income tax=3tr*0.05
Ti= 8tr                 income tax=5tr*0.05+(8tr-5tr)*0.1
Ti=12tr         income tax= 5tr*0.05+5tr*0.1+(12tr-10)*0.15
Ti=19tr         income tax=5tr*0.05+5tr*0.1+8tr*0.15+(19tr-18tr)*0.2
```

Write a program which will compute income tax of a people using the following interface:

Case 1:

```
Your income of this year: 240000000
Number of dependent:4
Tax-free income: 280800000
Taxable income: 0
Income tax: 0
```

Case 1:

```
Your income of this year: 440000000
Number of dependent:4
Tax-free income: 280800000
Taxable income:: 159200000
Income tax: 30190000
```

Program 3 (1 mark)

Objectives	Practice loop statements
Related knowledge	None
Problem	Write a C program that will print out **sum** of **integers** inputted from the keyboard until the value 0 is inputted.
Analysis *Nouns: sum → int S;* *Accepted integral value → int x*	**Suggested algorithm (logical order of verbs)** Begin S=0; Do { Accept x; //scanf("%d",&x); If (x != 0) S = S + x; } While (x!=0); Print out S; End

Program 4 (1 mark)

Objectives	Practice loops statement
Related knowledge	None
Problem	Write a C program that will carry out some times: accept two integers, swap these values, print them out to the monitor. The program will terminate when the value of 0 is inputted.
Analysis *Nouns:* *2 integers → int x, y;*	**Suggested algorithm (logical order of verbs)** Begin Do { Accept x, y; int t= x; /* t: temporary variable */ x= y; y= t; Print out x, y; } While (x!=0 && y!=0); End

Program 5: (2 marks)

Related knowledge	Use the function **getchar()** –stdio.h**,** to input a character, the function **toupper(ch)** to convert a character to uppercase - **ctype.h** ASCII code of the ENTER key: '\n'
Problem	Write a C program that will: - permit user inputting a string of characters. The input operation will terminate if the ENTER key is stroked. - print out the number of vowels, number of consonants, and number of others to the monitor.
Analysis *Nouns:* *inputted character* *→ char ch* *Number of vowels* *→ int nVowels =0;* *Number of consonants* *→ int consonants =0;* *Number of other characters → int nOthers =0;*	**Suggested algorithm (logical order of verbs)** Begin Do { Accept ch; /* ch= getchar(); */ Convert ch to its uppercase /* ch= toupper(ch); */ If (ch>='A' and ch <='Z') { switch (ch) { case 'A' : case 'E' : case 'I' : case 'O' : case 'U' : nVowels ++; break; default: nConsonants++; } } else nOthers = nOthers++; } While (ch != '\n'); Print out nVowels; Print out nConsonants; Print out nOthers; End

Program 6: (1 marks)

Related knowledge	Each character will be stored as its ASCII code with value 0..255
Problem	Write a C program that will print out the ASCII code table.
Analysis ASCII code	**Suggested algorithm (logical order of verbs)** Begin For each code = 0 to 255

→ int code	{ Print out ("%c : %d, %o, %X\n", code, code, code, code); If (code !=0 && code %20==0) getchar(); **/* code page of 20 lines */** } End.

Program 7: (1 marks)

Problem	Write a C program that will accept two characters then print out ASCII code difference between them and characters between them including code values in decimal, octal, hexadecimal expansions in ascending order.
Analysis 2 character → char c1, c2 Difference → int d; Character for swapping operation → char t Character for looping ➔ Char c	**Suggested algorithm (logical order of verbs)** Begin Accept c1 ;; Accept c2; If (c1 > c2) { t = c1; c1 = c2; c2= t; } d = c2 – c1; Print out d; For each c from c1 to c2 { Print out ("%c : %d, %o, %X\n", c, c, c, c); } End.

END