**B - Computation**

**Theory:**

1. **Variable**

* A variable is a name reference to a memory location, holds data that can change in value during the lifetime of the variable.

* The C language associates a data type with each variable.  Each data type occupies a compiler-defined number of bytes.

1. **Data Type**

Type 
char 
unsigned char 
int 
unsigned int 
short 
unsigned short 
long long 
unsigned long 
long 
float 
double 
Size 
1 byte 
1 byte 
4bytes 
4bytes 
2bytes 
2bytes 
8bytes 
8bytes 
4bytes 
8bytes 
Range 
-127 t0127 
O to 255 
-2147483648 to 
2147483647 
O to 4294967295 
-32768 to 32767 
O to 65,535 
-(263) to (263) - 1 
0 to 
18446744073709551615 

1. **Format string for Input/Output**

Type 
int 
char 
float 
double 
short int 
unsigned int 
long int 
long long int 
unsigned long int 
unsigned long long int 
signed char 
unsigned char 
long double 
Size (bytes) 
at least 2, usually 4 
4 
8 
2 usually 
at least 2, usually 4 
at least 4, usually 8 
at least 8 
at least 4 
at least 8 
at least 10, usually 12 or 16 
Format Specifier 
%f 
%hd 
%llu 
%Lf 
0/04 
0/01 i 

**Practice 1**: Write a program to create 2 integer variables a and b, assign value 254 to a and 343 to b then display the following result on the screen:



Where {P} is the sum of a and b.

----

Program:

Practicel .c 
3 
#include<stdio.h> 
int main 
printf( "a + b = %d 
return 0; 

----------

**Note**:

Variable naming conventions:

1. A valid identifier can have letters (both uppercase and lowercase letters), digits and underscores.
2. The first letter of an identifier should be either a letter or an underscore.
3. You cannot use keywords like int, while etc. as identifiers.
4. There is no rule on how long an identifier can be. However, you may run into problems in some compilers if the identifier is longer than 31 characters.

**Practice 2**: Write a program to declare 2 variables a and b which are real numbers, assign value 10.5 to a, 7 to b and display the following line on the screen:



where {P} is the quotient of a and b.

-----

Program:

Practice2.c 
1 
2 
3 
4 
5 
6 
#include<stdio.h> 
int main 
return 
o; 

**Practice 3**: Given a rectangle with the length of 7.5 and the width of 3.8. Write a program to print the area of this rectangle on the screen as below:

Area 
Ⅱ 宀 P} 

Where {P} is the area of the rectangle.

-----

Program:

Practice3,c 
1 
2 
3 
4 
5 
6 
#include<stdio.h> 
int main 
return 
o; 

**Practice 4**: Write a program to declare a character variable c, assign value 'x' to c and display variable c on the screen.

-----

Program:

Practice4.c 
1 
2 
3 
4 
5 
6 
7 
#include<stdio.h> 
int main 
printf( %c 
return 0; 

**Practice 5:** Write a program to display character 'd' on the screen.

-----

Program:

Practice5.c 
1 
2 
3 
4 
5 
6 
7 
#include<stdio.h> 
int main 
char d = 
printf( " , 
return ê', 

**Practice 6**: Write a program to declare two integers variables a and b, assign 384847522 to a, 988347273 to b and display the product of a and b on the screen.

-----

Program:

Practice6.c 
1 
2 
3 
4 
5 
6 
#include<stdio.h> 
int main 
return 
o; 

---------

Note:

In C programming language, there are integer data types such as: short, int, long, long long...

The difference among these data types are:

1. The range of type short is from -32768 to 32767.
2. The range of type int is from -2147483648to 2147483647.
3. The range of type long is from -2147483647to 2147483647 (the same as the rage of int).
4. The range of type long long is from -9223372036854775808 to 9223372036854775807

**Practice 7**: Write a program to read from keyboard the width and the height of a rectangle (the width and the height of this rectangle are integers) then display the following line on the screen:



where {P} is the area of the rectangle.

-----

Program:

Practice7.c 
1 
2 
3 
4 
5 
6 
#include<stdio.h> 
int main 
return 
o; 

**Practice 8**: Given two integers a and b. Write a program that accepts the values of a and b then displays the following line on the screen:



where {P} is the remainder when we divide a by b.

For example, if you enter two integers:



the screen will display:

a mod b = 
1 

Explanation: 7 modulo 3 is equal to 1

---

Program:

Practice8.c 
1 
2 
3 
4 
5 
6 
#include<stdio.h> 
int main 
return 
o; 

**Practice 9**: Given two integer values a and b. Write a program that accepts a and b and swap the value of a and b. Then print the following line on the screen:

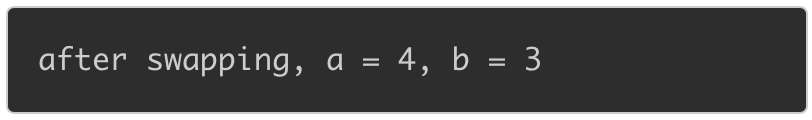
after swapping, a = 
{PI}, b 

where {P1} is the value of a after swapping and {P2} is the value of b after swapping.

For example, if a = 3 and b = 4, we enter the following values:



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



-----

Program:

Practiceg.c 
b) 
a- 
3 
#include<stdio.h> 
int main 
int a, b; 
scanf( " , 
printf( "after 
return ê', 
swapping, 
%d, 
a 

**EXERCISE**

**EX1**: Write a C program that prompts the user to enter the amount of cash in their pockets, accepts the user's input, and displays the amount in the format shown below.

If the user has entered 4.52, your program displays

How much money do you have in your pockets ? 4.52 
The amount of money in your pockets is $4 .52 

**EX2**: Write a C program that calculates the area of a triangle.  Your program prompts the user to enter the height and base, accepts the user's input in decimal format, multiplies the product of the height and base by 0.5, stores the area in memory and outputs the area to 2 decimal places along with its address in memory in hexadecimal format (use the **%p** conversion specifier to output an address).