

Embedded C Programming – Laboratory (L33+L34)

Experiment 1 – Programs on Sequential Statements

Task 1.1:

```
#include<stdio.h>

int main(){
    printf("Hello World");
    return 0; //Return 0 to indicate successful execution
}
```

Output:

```
Hello World
-----
Process exited after 0.01559 seconds with return value 0
Press any key to continue . . .
```

Task 1.2:

```
#include <stdio.h>

int main() {
    // Sequential statements
    printf("This is a C program.\n"); // Statement 1
    printf("It consists of sequential statements.\n"); //
Statement 2
    printf("Each statement is executed one after the
other.\n"); // Statement 3
    return 0;
}
```

Output:

```
This is a C program.
It consists of sequential statements.
Each statement is executed one after the other.
-----
Process exited after 0.01053 seconds with return value 0
Press any key to continue . . . |
```

Task 1.3:

```
#include <stdio.h>

int main() {
    int x = 5, y = 3;
    printf("%d", x + y);
    return 0;
}
```

Output:

```
8
-----
Process exited after 0.01375 seconds with return value 0
Press any key to continue . . .
```

Task 1.4a:

```
main(){
    /*printing begins*/
    printf("I see, I remember");
    /*printing ends*/
}
```

Output:

```
I see, I remember
-----
Process exited after 0.01247 seconds with return value 17
Press any key to continue . . . |
```

Message:

Compiler (2) Resources Compile Log Debug Find Results Close			Message
Line	Col	File	In function 'main':
3	2	D:\OneDrive - vit.ac.in\My Classes\Fall_2024-Embedd...	[Warning] incompatible implicit declaration of built-in function 'printf'

Task 1.4b:

```
#include<stdio.h>

main(){

    /*printing begins*/

    printf("I see, I remember");

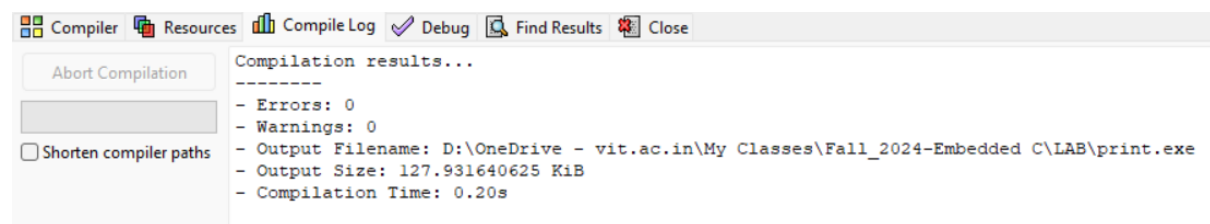
    /*printing ends*/

}
```

Output:

```
I see, I remember
-----
Process exited after 0.01247 seconds with return value 17
Press any key to continue . . . |
```

Message:



Task 1.4c:

```
#include<stdio.h>

main(){

    /*printing begins*/

    printf("This is Embedded C Programming, \nExperiment
No. 1");

    /*printing ends*/

}
```

Output:

```
This is Embedded C Programming,
Experiment No. 1
-----
Process exited after 0.01818 seconds with return value 49
Press any key to continue . . . |
```

Task 1.5:

```
/* Programm ADDITION*/  
#include<stdio.h>  
main()  
{  
    int number;  
    float amount;  
    number = 100;  
    amount = 30.75 + 75.35;  
    printf("%d\n",number);  
    printf("%f",amount);  
}
```

Output:

```
100  
106.099998  
-----  
Process exited after 0.01578 seconds with return value 10  
Press any key to continue . . . |
```

Task 1.6:

```
/* Programm ADDITION*/  
#include<stdio.h>  
main()  
{  
    int number;  
    float amount;  
    number = 100;  
    amount = 30.75 + 75.35;  
    printf("%d\n",number);
```

```
    printf("%.3f",amount);  
}
```

Output:

```
100  
106.100  
-----  
Process exited after 0.01329 seconds with return value 7  
Press any key to continue . . . |
```

Task 1.7:

```
#include<stdio.h>  
  
int main() { // Sequential statements  
    int a = 5;        // Statement 1: Variable  
    declaration and initialization  
    int b = 3;        // Statement 2: Variable  
    declaration and initialization  
    int sum = a + b;   // Statement 3: Addition operation  
    int product = a * b; // Statement 4: Multiplication  
    operation  
    // Displaying results  
    printf("The sum of %d and %d is: %d\n", a, b, sum); //  
    Statement 5: Output  
    printf("The product of %d and %d is: %d\n", a, b,  
    product); // Statement 6: Output  
    return 0;  
}
```

Output:

```
The sum of 5 and 3 is: 8  
The product of 5 and 3 is: 15  
-----  
Process exited after 0.01066 seconds with return value 0  
Press any key to continue . . .
```

Task 1.8:

```
#include<stdio.h>

int main() { // Sequential statements

    int a = 20;    // Statement 1: Variable declaration
and initialization

    int b = 4;     // Statement 2: Variable declaration
and initialization

    int result_subtraction = a - b; // Statement 3:
Subtraction operation

    int result_division = a / b; // Statement 4: Division
operation

    // Displaying results

    printf("The result of %d - %d is: %d\n", a, b,
result_subtraction); // Statement 5: Output

    printf("The result of %d / %d is: %d\n", a, b,
result_division);    // Statement 6: Output

    return 0;

}
```

Output:

```
The result of 20 - 4 is: 16
The result of 20 / 4 is: 5

-----
Process exited after 0.01104 seconds with return value 0
Press any key to continue . . . |
```

Task 1.9:

```
#include <stdio.h>

int main() {

    // Sequential statements with user input

    int a, b;

    printf("Enter the value of a: ");
```

```

scanf("%d", &a); // User input for 'a'
printf("Enter the value of b: ");
scanf("%d", &b); // User input for 'b'
int result_subtraction = a - b;
int result_division = a / b;
// Displaying results
printf("The result of %d - %d is: %d\n", a, b,
result_subtraction);
printf("The result of %d / %d is: %d\n", a, b,
result_division);
return 0;
}

```

Output:

```

Enter the value of a: 3
Enter the value of b: 21
The result of 3 - 21 is: -18
The result of 3 / 21 is: 0

-----
Process exited after 11.47 seconds with return value 0
Press any key to continue . . .

```

Task 1.10:

```

#include <stdio.h>

int main() {
    int a, b;
    printf("Enter the value of a: ");
    scanf("%d", &a); // User input for 'a'
    printf("Enter the value of b: ");
    scanf("%d", &b); // User input for 'b'
    // Division operation with quotient and remainder

```

```

        int quotient = a / b;    // Statement 4: Division
operation for quotient

        int remainder = a % b;  // Statement 5: Modulo
operation for remainder

    // Displaying results

    printf("The result of %d / %d is quotient: %d,
remainder: %d\n", a, b, quotient, remainder); //
Statement 6: Output

    return 0;

}

```

Output:

```

Enter the value of a: 3
Enter the value of b: 21
The result of 3 / 21 is quotient: 0, remainder: 3

-----
Process exited after 6.919 seconds with return value 0
Press any key to continue . . .

```

```

Enter the value of a: 21
Enter the value of b: 3
The result of 21 / 3 is quotient: 7, remainder: 0

-----
Process exited after 4.145 seconds with return value 0
Press any key to continue . . . |

```

Task 1.11:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main() {
```

```
    // Using char data type to store user input
```

```
    char userInput;
```



```

    printf("Enter a number: ");
    scanf("%c", &userInput);

    // Converting the character to a numeric value
    (assuming ASCII representation)
    // and calculating the square root
    int numericValue = userInput - '0';
    double squareRootResult = sqrt(numericValue);

    // Displaying the result
    printf("Square root of %d is: %.2f\n", numericValue,
squareRootResult);

    return 0;
}

```

Output:

```

Enter a number: k
Square root of 59 is: 7.68

-----
Process exited after 10.59 seconds with return value 0
Press any key to continue . . .

```

Task 1.12:

```

#include <stdio.h>

//Example on datatypes and format specifiers - sequential
int main() {
    // int
    int integerVariable = 42;
    printf("int: %d\n", integerVariable);
}

```

```
// float
float floatVariable = 3.14;
printf("float: %.2f\n", floatVariable);

// double
double doubleVariable = 3.14;
printf("double: %.2f\n", doubleVariable);

// char
char charVariable = 'A';
printf("char: %c\n", charVariable);

// _Bool
_Bool boolVariable = 1; // true
printf("_Bool: %d\n", boolVariable);

// short
short shortVariable = 32767; // can use either short or
short int
printf("short: %d\n", shortVariable);
printf("The size of short int is: %zu bytes",
sizeof(shortVariable));

// long
long longVariable = 2147483647; // can use either long
or long int
printf("long: %ld\n", longVariable);
```

```
// unsigned int
unsigned int positiveInteger = 42;//can use either
long or long int

printf("unsigned int: %u\n", positiveInteger);

return 0;
}
```

Output:

```
int: 42
float: 3.14
double: 3.14
char: A
_Bool: 1
short: 32767
The size of short int is: 2 byteslong: 2147483647
unsigned int: 42

-----
Process exited after 7.317 seconds with return value 0
Press any key to continue . . .
```