

Q. Why and when do we use the #define directive?

→ In the C programming language, the #define directive allows the definition of macros within our source code. These macro definitions allow constant values to be declared for using in our code. Macro definitions are not variables and cannot be changed by our program code like variables. We generally use this syntax when creating constants that represents numbers, strings or expressions. The syntax of using #define in the C language is:

#define CNAME value

OR

#define CNAME (expression)

Here, CNAME contains the name of the constant. Most C programmers define their constants names in uppercase. value contains the value of the constant. Here, expression means whose value is assigned to the constant. The expression must be enclosed in parentheses if it contains operators.

Example: #define AGE 10

#define AGE (20/2)



Sub: \_\_\_\_\_

Day

Time: \_\_\_\_\_

Date: / /

→ Why and when do we use the #include directive?

→ In the C programming language, the #include directive tells the preprocessor to insert the contents of the another file into the source code. Include directive are typically used to include the C header files for C functions that are held outside of the current source file.

The syntax for the #include directive in the C language is:

#include <headerfile>  
OR  
#include "headerfile"

Here, (headerfile) contains the name of the header file that we wish to include. A header file is a C file that typically ends in ".h".

Example:

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

```
int main()
```

```
{  
    int a, b; a = 10, b = 20;
```

```
    int sum = a + b;
```

```
    printf("The result is %d", sum);
```

```
    return 0;
```

```
}
```

In this program, we are using the #include



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

directive to include the `stdio.h` header file which is required to use the `printf` and `scanf()` standard C library function in our code.

Q1 What does `void main(void)` mean?

→ Here, the `void main()` indicates that `main()` function will not return any value. The `void` in parenthesis (`void`) means that the `main` function accepts no arguments.

When our program is simple and it is not going to terminate before reaching the last line of the code or the code is error that we can use the `void main()`.

Q2 What do we need to use comments in Programs?

→ In computer programming language, a comment is a programmer-readable explanation in the source code of a computer program. They are added with the purpose of making the source code easier for humans to understand. And generally ignored by the compilers and the interpreters.

A comments starts with a slash asterisk (`/*`) and ends with a asterisk slash (`*/`) and it can be anywhere in our program. comments can span several lines within our C program.

The syntax of a comment is:

`/* comment goes here */`



Sub: \_\_\_\_\_

Day: \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

Example: `/* This is  
a multiline  
comment */`

Comments are also be in single lines.

Example: `// this is a single line comment.`

[single line comments are started with double slash.]

Write down the basic structure of C program.

→ To write a C program, it may contain one or more sections shown in figure.

Documentation Section

Link Section (#include section)

Definition Section

Global declaration section

main() function section

Declaration Part;

execution Part;

subprogram section

`/* comment goes here */`

Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

Documentation Section: The documentation section is the part of the program where the programmer gives the details associated with the program.

Link Section: This part of the program/code is used to declare all the header files that will be used in the program.

Definition Section: In this section, we define different constants. Like `#define PI 3.1416`

Global Variable Selection: In this part, the global variables are declared. Like `int a = 7;`

Main Function: Every C program needs to have the main function. Each main function contains 2 parts. A declaration part and an execution part.

Subprogram Section: All the user defined function are defined in this section of the program.



Sub: \_\_\_\_\_

Day

Time: \_\_\_\_\_

Date: / /

Q1. How many steps are there to write a C program?

Ans. Executing a C program.

→ Executing a program written in C involves a series of steps. These are:

1. Creating the program
2. Compiling the program
3. Linking the program with functions
4. Executing the program

goto next  
page

Subprogram section: All the user defined functions are defined in this section of the program.

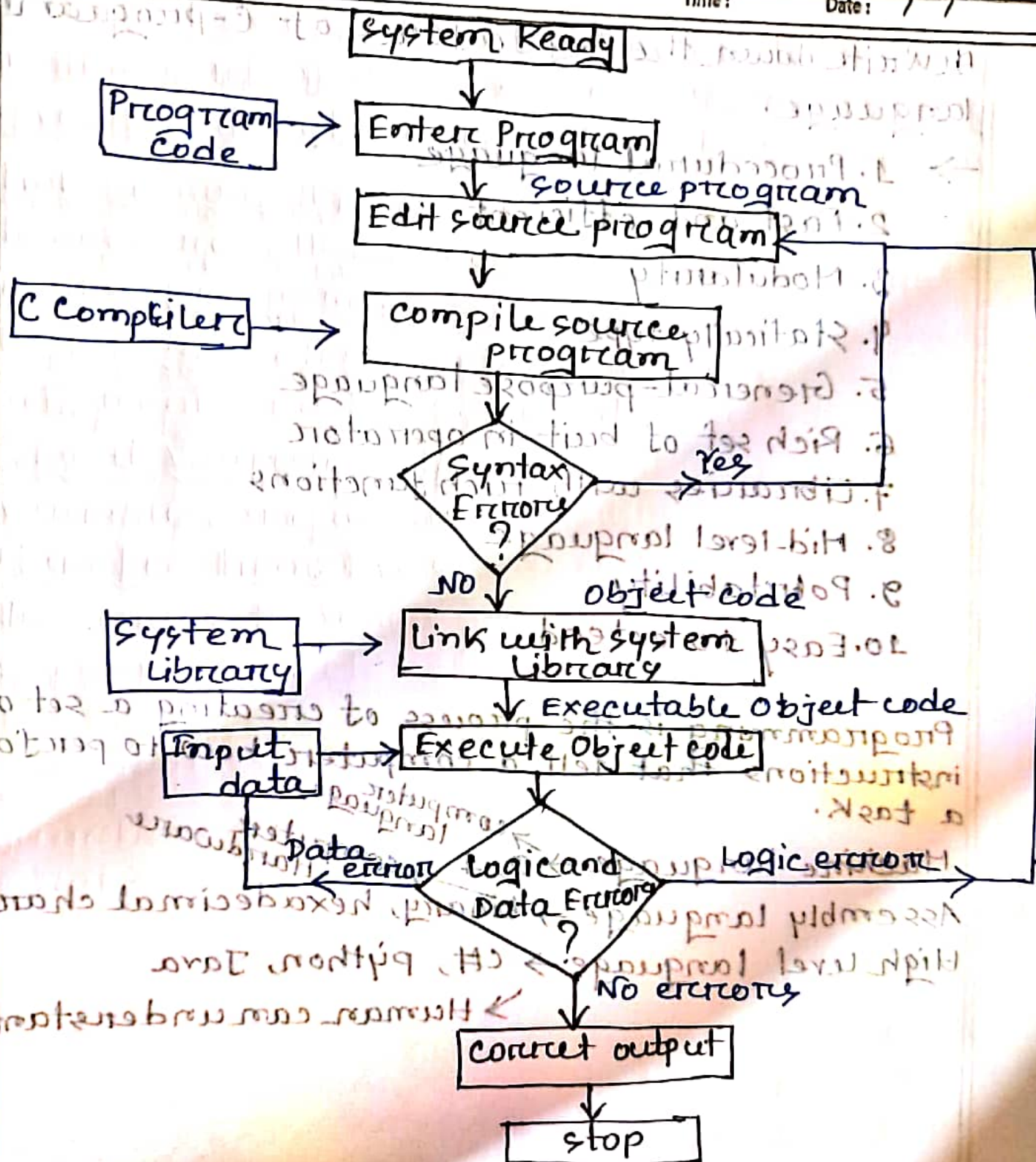


Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /



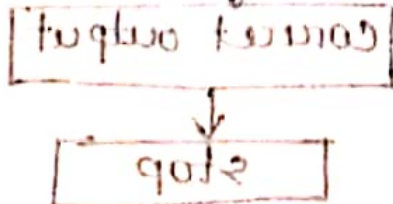
Process of compiling and running a C program.

Write down the key features of C-programming language.

- 
1. Procedural language
  2. Fast and efficient
  3. Modularity
  4. Statically type
  5. General-purpose language
  6. Rich set of built-in operators
  7. Libraries with rich functions
  8. Mid-level language
  9. Portability
  10. Easy to extend

Programming is the process of creating a set of instructions that tell a computer how to perform a task.

Machine language → computer language  
Assembly language → binary, hexadecimal characters  
High level language → C#, Python, Java  
→ Human can understand





Sub: \_\_\_\_\_

Day: \_\_\_\_\_  
 Time: \_\_\_\_\_ Date: \_\_\_\_/\_\_\_\_/\_\_\_\_

Q1 Write down the Character Test Functions  
 → Some character test functions are following in table:

Function	Test
isalnum(c)	Is c an alphanumeric character?
isalpha(c)	Is c an alphabetic character?
isdigit(c)	Is c a digit?
islower(c)	Is c lower case letter?
isprint(c)	Is c a printable character?
ispunct(c)	Is c a punctuation mark?
isspace(c)	Is c a white space character?
isupper(c)	Is c a uppercase letter?

Q2 What is Formatted Input? Explain.

→ In C programming, scanf() is one of the commonly used function to take input from the user. The scanf() function reads formatted input such as keyboards.

Formatted input function scanf(), is used to take various inputs from the user. This type of function can help to display the output to the user in different formats using the format specifiers. Formatted input function support all data types like int, float, char, double and many more.



Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

For example:

15.75 123 John

Here, this line contains three pieces of data, arranged in a particular form. The first part of the data should be read into a variable float, the second into int and the third part into char. This is possible in C using the scanf function. scanf means scan formatted.

scanf() function is used in C program for reading or taking any value from the keyboard by the user. This function is declared in stdio.h (header file), that's why it is also predefined function. In scanf() function, we use address of operator (&) which is used to store the variable value on the memory location of that variable. The general

format of scanf() is

scanf("format specifier", &var1, &var2, ..., &varn);

scanf("control string", &arg1, &arg2, ..., &argn);



example:

```
#include <stdio.h>

int main()
{
    int a;
    float b;
    printf("Enter integer and then a float: ");
    scanf("%d %f", &a, &b); // Taking multiple inputs
    printf("You entered %d and %f", a, b);
    return 0;
}
```

Output: Enter integer and then a float: 3  
2.4  
You entered 3 and 3.400000

3.4.11

When searching for a value, scanf uses the next appropriate character. When the field width specifier is used, it should be large enough to contain the input data size.



Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

Some Points to remember while using `scanf()`  
→ Some of the general points to keep in mind while writing a `scanf()` statement:

1. All function arguments, except the control string must be pointers to variables.
2. Format specifiers contained in the control string, should match the arguments in order.
3. Input data items must be separated by spaces and must match the variables receiving the input in the same order.
4. The reading will be terminated, when `scanf` encounters a mismatch of data of a character that is not valid for the value being read.
5. When searching for a value, `scanf` ignores line boundaries and simply looks for the next appropriate character.
6. When the field width specifier `w` is used, it should be large enough to contain the input data size.



Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

## Rules For scanf()

- Each variable to be read must have a format specifier.
- For each format specifier, there must be a variable address of proper type.
- Any non-whitespace character used in the format string must have matching in the user input.
- The scanf reads until:
  - A white space character is found in a numeric specification.
  - The maximum number of characters have been read or
  - An error is detected or
  - The end of file is reached.

Example:

```
int a = 20;
float b = 30.05;
printf("a = %d, b = %f", a, b);
return 0;
```

Output: 20 30.050000



Sub : \_\_\_\_\_

Day \_\_\_\_\_

Time : \_\_\_\_\_

Date : / /

Q1 What is formatted output? explain.

→ Formatted

→ The C language comes with standard function `printf()` so that a programmer can perform formatted output in a program.

This type of function can help to display the output to the user in different format using the format specifiers. This function supports all data types like `int`, `float`, `char`, `double` and many more.

`printf()` function is used in a C program to display any value like `float`, `integer`, `character`, `string` etc on the screen. It is a predefined function that is already declared in the `stdio.h` (header file).

The syntax of `printf()` function is :

`printf("format specifier", var1, var2, ..., varn);`

Example:

```
int main()
```

```
{ int a=20;
```

```
  float b = 30.05;
```

```
  printf("%d %f", a, b); // display multiple output
```

```
  return 0;
```

```
}
```

output : 20 30.05000000

`printf("control string", arg1, arg2, ..., argn);`



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

The control string consists of following three types of items:

1. Characters that will be printed on the screen as they appear.

2. Format specifications that define the output format for display of each item.

3. Escape sequence characters such as `\n`, `\t` and `\b`.

`printf` never supplies a newline automatically. A newline or space can be introduced by the help of a newline character `"\n"`.

```
printf("\n");
```

```
printf(" ");
```

```
printf("\n\n");
```

Example:

```
printf("a = %d\t b = %d", a, b);
```

```
printf("a = %d\n b = %d", a, b);
```

```
printf("code / Name of file\n");
```

```
printf("Enter your name\n");
```



Sub: \_\_\_\_\_

Day \_\_\_\_\_

Time: \_\_\_\_\_

Date: / /

#### 4. Enhancing the readability of output

→ Readability of output is very important.

The correctness depends on the solution procedure, the clarity depends on the way the output is presented. Following are some of steps that we can take to improve the clarity and hence the readability and understandability of outputs:

1. Provide enough blank space between two numbers.
2. Introduce appropriate headings and variable names in the output.
3. Print special messages whenever a special condition occurs in the output.
4. Introduce blank lines between the important sections of the output.

Example:

```
printf("a = %.d\t b = %.d", a, b);  
printf("a = %.d\n b = %.d", a, b);  
printf("code\t Name\t Age\n");  
printf("Enter your name\n");
```



Sub: \_\_\_\_\_

Day

--	--	--	--	--	--	--	--

Time: \_\_\_\_\_

Date: / /

Q. How can we use the `getchar()` function to read multiple-character strings?

→ All input or output operations is reading a character from the 'standard input' unit (usually the keyboard) and writing it to the 'standard output' unit usually the screen. Reading a single character can be done by using the `getchar()`. The `getchar()` takes following form:

`variable-name = getchar();`

variable-name is a valid C name that has been declared as char type. When this statement is encountered, the computer waits until a key is pressed and then assign the character as a value to `getchar()`.

For example,

`char name;`

`name = getchar();`



Sub: \_\_\_\_\_

Day

Time: \_\_\_\_\_

Date: / /

4. How can we use the `putchar()` function to output multicharacter strings?

→ All input or output operations are done in character from the standard input/output (stdin/stdout) and writing it to the standard output. Reading a character can be done by using the `getchar()` function.

Example: `putchar('A');`

Variable `name` is a valid C name that has been declared as char type. When this statement is encountered, the computer will wait until a key is pressed and then assign the character as a value to `putchar()`.

char name;

`name = getchar();`