



SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Coimbatore – 641 035.



B.E / B. Tech – Internal Assessment Exam – I

Academic Year 2022 – 2023 (ODD)

FIRST SEMESTER (REGULATION R2019)

19CST101 – PROGRAMMING FOR PROBLEM SOLVING

ANSWER KEY

PART A

B

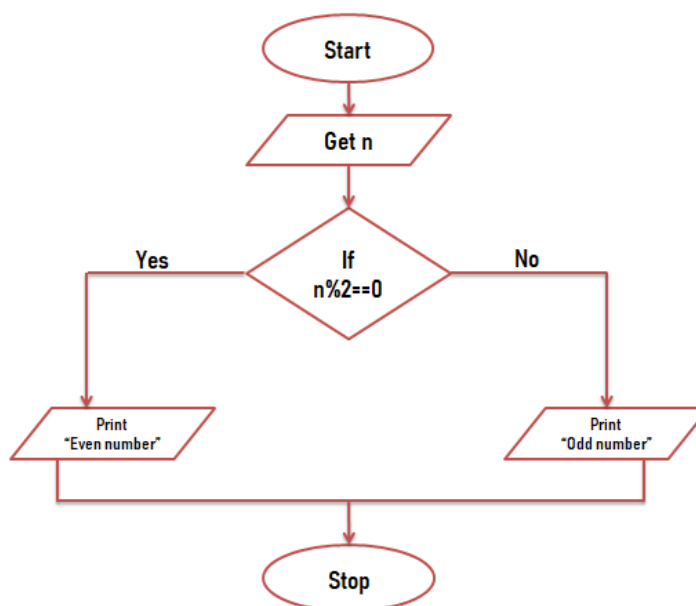
1. Write down the guidelines for writing an algorithm.

Ans: Guidelines for writing an Algorithm:

- ✍ Algorithm Should be written in simple English
- ✍ Each and every instruction should be precise and unambiguous.
- ✍ Instructions in an algorithm should not be repeated infinitely.
- ✍ Algorithm should conclude after a finite number of steps.
- ✍ Algorithm Should have an end point
- ✍ Derived results should be obtained only after the algorithm terminates

2. Draw the flow chart to find given number is Even or Odd.

Ans: Flowchart to find whether the given number is Even or Odd:



3. Figure out the pseudocode for conversion of Fahrenheit to Celsius?

Ans: Pseudocode for the conversion of Fahrenheit to Celsius

BEGIN

DECLARE FAHRENHEIT, CELSIUS

GET FAHRENHEIT

SET CELSIUS = $5 * (\text{FAHRENHEIT} - 32) / 9$

DISPLAY CELSIUS

END

4. List out keywords used in C.

Ans: Keywords in C

auto, double, int, struct, break, long, switch, if, else, case, enum, register, typedef, char, extern, return, union, const, float, short, unsigned, continue, for, signed, void, default, goto, sizeof, volatile, do, static, while

5. Write a C program to print college name.

Ans:







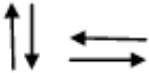
```
#include<stdio.h>

void main()
{
clrscr();
printf("SNS College of Technology");
getch();
}
```

PART B

6. a) Illustrate the symbols used in flowchart and guidelines for representing flowchart.

Ans: Symbols used in flowchart

Symbol	Name	Function
	Process	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Terminal	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

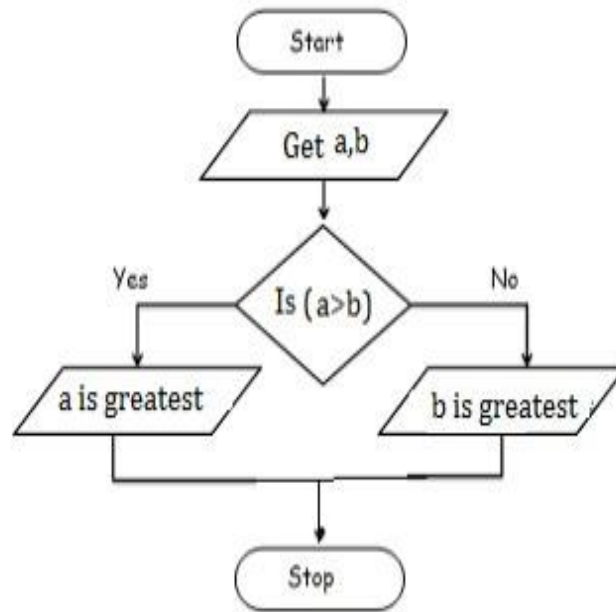
Guidelines for representing the Flowchart:

1. In drawing a proper flowchart, all necessary requirements should be listed out in logical order.
2. Flow chart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flowchart.
3. The usual directions of the flow of a procedure or system are from left to right or top to bottom.
4. Only one flow line should come out from a process symbol.
5. Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, can leave the decision symbol.
6. Only one flow line is used in conjunction with terminal symbol.

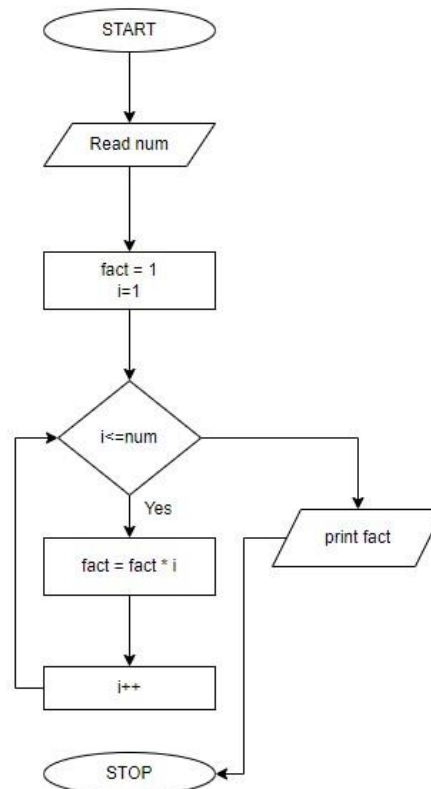
7. If flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines.
8. Ensure that flowchart has logical start and stop.

b) Draw flowchart to find i) Greatest of two numbers ii) Factorial using recursion.

Ans: i) Greatest of two numbers



ii) Factorial using Recursion



7. a) Enumerate the Structure of C programming with a Sample C program

Ans: Structure of C Program

Basically the structure of a C program divides into 6 sections and they are,

- ✍ Documentation
- ✍ Preprocessor
- ✍ Definition
- ✍ Global Declaration
- ✍ Main function
- ✍ User defined functions

Documentation Section

- ✓ It includes the statement specified at the beginning of a program, such as a program's **name, date, description, and title**, which is represented using commands
- ✓ Single line commands will be represented by //
- ✓ Multi – line commands will be represented as /* */

Eg: //program1.c

Preprocessor Section

- ✓ The preprocessor section contains all the header files used in a program. It informs the system to link the header files to the system libraries
- ✓ A header file in C/C++ contains:
 - i) **Function definitions** ii) **Data type definitions** iii) **Macros**
- ✓ Header files offer these features by importing them into your program with the help of a preprocessor directive called **#include**.
- ✓ These preprocessor directives are responsible for instructing the C/C++ compiler that these files need to be processed before compilation.
- ✓ Every C program should necessarily contain the header file **<stdio.h>** which stands for standard input and output used to take input with the help of **scanf() function** and display the output using **printf() function**.
- ✓ The source file contains **#include** which is responsible for directing the C/C++ compiler that this file needs to be processed before compilation and includes all the necessary data type and function definitions

Define Section

The define section comprises of different constants declared using the define keyword

Eg: #define a = 4

Global Declaration

- ✍ The global section comprises of all the global declarations in the program.
- ✍ Anything which is declared as global can be used throughout the entire program
- ✍ It should be declared before the main function

Main Function

- ✍ main() is the first function to be executed by the computer.
- ✍ It is necessary for a code to include the main(). It is like any other function available in the C library.
- ✍ Parenthesis () are used for passing parameters (if any) to a function.

User defined functions

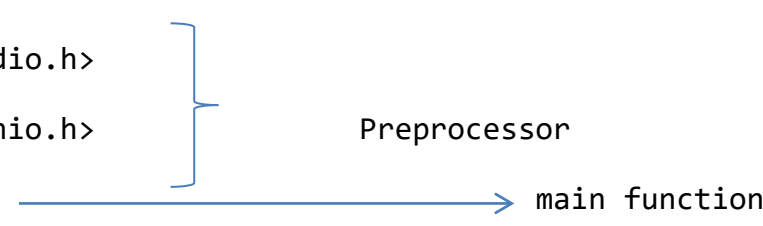
- ✍ The user defined functions specified the functions specified as per the requirements of the user.
- ✍ For example, color(), sum(), division(), etc.

Basic Syntax of a C program

```
//Documentation Section  
  
#include<stdio.h> //Preprocessor section  
  
//Define section  
  
//Global Declaration  
  
void main //Main function  
{  
  
//Statements to be executed  
  
//User defined functions (if any..)  
}
```

Example

```
//Program to find the factorial of a number  -- Documentation
#include<stdio.h>
#include<conio.h>
void main()
{
int n, fact=1, i;
printf("Enter the value for n");
scanf("%d", &n);
for(i=1; i<=n; i++)
{
fact = fact * i;
}
printf("\n The Factorial of %d is %d", n, fact);
}
```



Output:

```
Enter the value for n: 5
```

```
Factorial of 5 is 125
```

b) Explain about C tokens and rules for identifiers.

Ans: C Tokens

C tokens, Identifiers and Keywords are the basic elements of a C program. C tokens are the basic buildings blocks in C. Smallest individual units in a C program are the C tokens.

C tokens are of six types. They are,

- ✍ Keywords (e.g.: int, while)
- ✍ Identifiers (e.g.: main, total)
- ✍ Constants (e.g.: 10, 20)

- ✍ Strings (e.g.: “total”, “hello”)
- ✍ Delimiters (e.g.: (), {})
- ✍ Operators (e.g.: +, /, -, *)

KEYWORDS

Keywords are those words whose meaning is already defined by Compiler. They cannot be used as Variable Names. There are 32 Keywords in C.

C Keywords are also called as **Reserved words**.

There are 32 keywords in C.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

IDENTIFIERS

Identifiers are the names given to various program elements such as variables, arrays & functions. Basically, identifiers are the sequences of alphabets or digits.

Rules for forming identifier name

The first character must be an alphabet (uppercase or lowercase) or an underscore.

All succeeding characters must be letters or digits.

No space and special symbols are allowed between the identifiers.

No two successive underscores are allowed.

Keywords shouldn't be used as identifiers.

CONSTANTS

The constants refer to fixed values that the program may not change or modify during its execution. Constants can be of any of the basic data types like an integer constant, a floating constant and a character constant. There is also a special type of constant called enumeration constant.

E.g.:

Integer Constants: 45, 215u

Floating Constants: 3.14, 4513E-5L

Character Constants: \t, \n

STRINGS

A string in C is actually a one-dimensional array of characters which is terminated by a null character '\0'.

E.g.:

```
char str = {'S', 'A', 'T', 'H', 'Y', 'A', 'B', 'A', 'M', 'A'}
```

DELIMITERS

The symbols other than alphabets, digits and white spaces for example - [] () { }

comma(,) Semicolon(;) colon(:) asterisk(*) ellipsis(...) equal to(=)

hash(#) are the special symbols.

8. a) Develop a Flowchart, Algorithm, Pseudocode for

i) To print given number is even / odd / zero

ii) Square of the number

Ans:

i) To print given number is even / odd / zero

ALGORITHM

- Step 1: Start
- Step 2: get num
- Step 3: check if(num==0) print num is zero
- Step 4: else if(num%2==0) print num is even
- Step 5: else num is odd
- Step 6: Stop

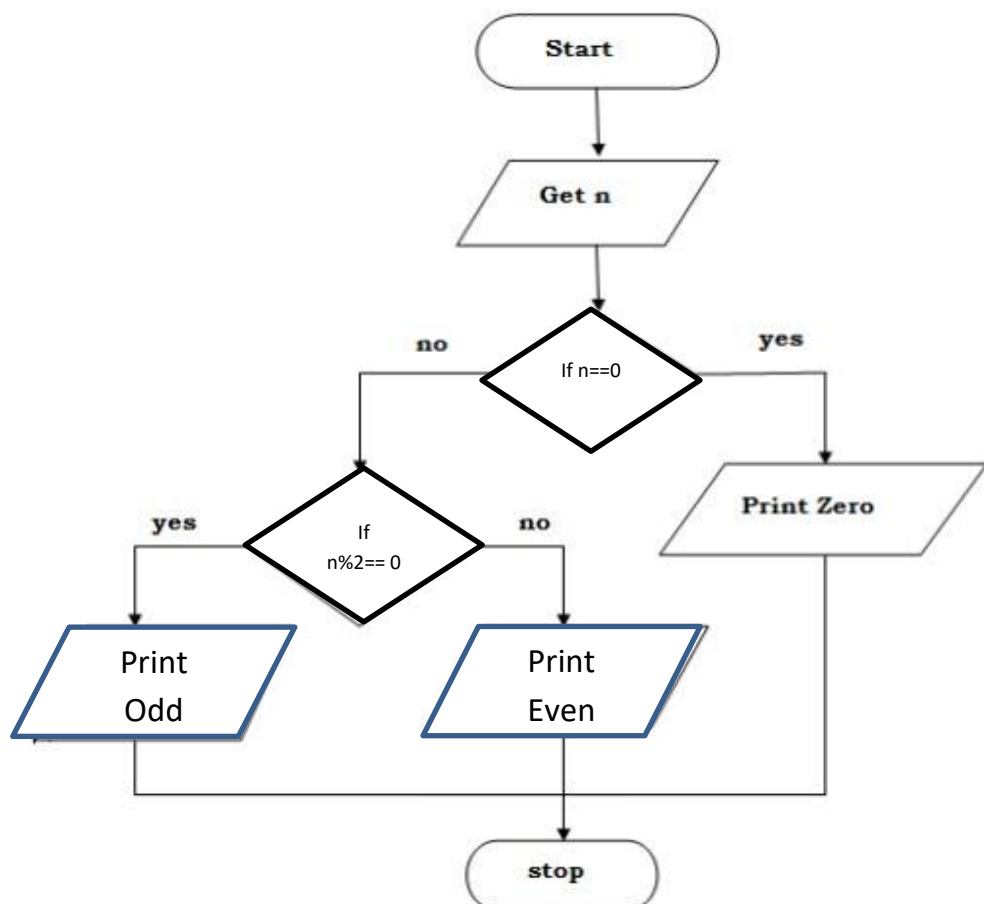
PSEUDOCODE

```

BEGIN
READ num
IF (num == 0) THEN
DISPLAY num is zero
ELSE IF (num%2==0) THEN
DISPLAY num is even
ELSE
DISPLAY num is odd
END IF
END

```

FLOWCHART



b) Compose the C program for:

i) Finding the area & circumference of a circle

```
#include<stdio.h>

#include<conio.h>

void main()

{

    float radius, area, circum;

    printf("\n Enter the radius of Circle : ");

    scanf("%d", &radius);

    area = 3.14 * radius * radius;

    circum = 2 * 3.14 * radius

    printf("\n Area of Circle : %f", area);

    printf("\n Circumference of Circle : %f", circum);

    return (0);

}
```

Output:

Enter radius of a circle : 1

Area of circle : 3.14

Circumference of circle : 6.28

ii) Calculate Engineering Cut off

```
#include <stdio.h>

void main()

{

int p, c, m, t, mp;

printf("Eligibility Criteria for an engineering:\n");
```

```

printf("Marks in Mathematics >= 65\n");
printf("Marks in Physics >= 55\n");
printf("Marks in Chemistry >= 50\n");
printf("Total in all three subject >= 180\n");
printf("or Total in Maths and Physics >= 140\n");
printf("-----\n");
printf("Input the marks obtained in Physics :");
scanf("%d", &p);
printf("Input the marks obtained in Chemistry :");
scanf("%d", &c);
printf("Input the marks obtained in Mathematics :");
scanf("%d", &m);
printf("Total marks of Mathematics, Physics and Chemistry : %d\n", m + p + c);
printf("Total marks of Maths and Physics : %d\n", m + p);
if (m>=65)
if(p>=55)
if(c>=50)
if((m + p + c) >= 180 || (m + p) >= 140)
printf("The candidate is eligible for admission.\n");
else
printf("The candidate is not eligible.\n");
else
printf("The candidate is not eligible.\n");
else
printf("The candidate is not eligible.\n");
else
printf("The candidate is not eligible.\n");
}

```