Fatima Ghaffar

"Programming Fundamentals"

What is Programming?

A set of instructions or a compound of words given to the computer to solve the problems, is called "Programming".

Computer:

Computer is an electronic (machine ~~device~~), that take the input from the user, processes it, and generate the output

Machine:

A machine is the device of any system, that can help/assit human beigs.

Types of Machine

A machine is of many types:

- Electronic machine
- Mechanical machine etc.

Electronic machine:

An electronic device is a system or machine that relies on electronic circuits, electric current (+ve or -ve charges) to process and control the information.

Why computer can only understand binary language?

Computer can only understand binary language because they are fundamentally designed to process information, using binary system, which use only two states, "0" and "1". These two states crosspends to the electrical signals in computer circuitry, typically off (0) and on (1).

Programming Languages

1	C++	15	JavaScript	29	PHP
2	CSS	16	Assembly language	30	Swift
3	D	17	Easytrieve Plus	31	Ada
4	R	18	Visual Basic	32	BASIC
5	Java	19	ActionScript	33	Objective-C
6	Python	20	Game Maker Language	34	Koltin
7	Go	21	TypeScript	35	ABAP
8	Ruby	22	Apache Groovy	36	ALGOL
9	Pascal	23	Transact-SQL	37	C#
10	Ballerind	24	VB.NET	38	SAS
11	Bash	25	NoSQL	39	Dart
12	MATLAB	26	HTML	40	Scheme
13	PL/SQL	27	Haskell	41	Lisp
14	Perl	28	COBOL	42	Scala

43	Delphi	46	Apex	49	Rust
44	Prolog	47	Haskell	50	Fortran
45	Lua	48	Julia	51	Scratch

Artificial Intelligence

↓ Intelligence is The ability to learn, understand and solve the problems.

Key functions

1. Learning ↓ Process to learn, something
2. Reasoning ↓ to justify something
3. Problem solving (God gifted) ↓ Use of alternate
   - Men have diff skil of problem solving
4. adopt ability ↓ to adopt something
   - company of humans
5. Understanding comes when interest remains
6. Decision Making to decide something
7. Memory to store something

KING'S NOTES

Date:

Artificial Intelligence

Intelligence is the ability to learn, understand and solve the problems.

Key functions:

Intelligence includes.

- Learning
- Reasoning
- Problem solving
- Adopt Ability
- Understanding
- Decision making
- Memory

1. Learning:

Learning is the process to learn, understand and memorize something.

This may includes experimental learning or theoretical learning.

Learning involves the observation and analysis to understand something.

2. Reasoning:

Reasoning refers to justify something Reasoning involves using logic or rational thinking to analyze and draw conclusions

Example:

Determining which route take to avoid traffic by analyzing road conditions and past experiences.

3. Problem solving: s

Men have different skills for problem solving; problem-solving varies between

KING'S NOTES

Date: M T W T F S S

individuals, The use of alternate methods refers to the ability to find creative or conventional solution of issues.

e.g: A mechanic fixing a car by diagnosing the issue and trying alternating methods when the intial fix does not work.

4. Adaptability:

It refers to adopt something, i.e: adjusting or adopting to new environment or circumstances.

e.g:

A student adapting to online classes during a pandemic by setting up a dedicated study space and learning to use virtual tools.

5. Understanding:

Understanding comes when interest remains, It implies that true understanding occurs when there is sustained curiosity or engangement.

e.g:

A teacher effectively explaining a complex math subject to a student who shows his interest and asks questions.

6. Decision Making:

Decision making is to decide something. It refers to choosing a course of action among several alternatives based on reasoning and understanding.

Date: M T W T F S S

e.g:

Choosing between two job offers by comparing salary, location and career growth opportunities.

7. **Memory:**

Memormy refers to store something. It involves retaining information for later retrieval or use, which is crucial for learning and problem solving.

e.g:

Remembering important dates, such as a friend's birthday, to maintain personal relationships.

Fatima Ghaffar

Reserved Words

1_ alignas
2_ alignof
3_ compl
4_ const
5_ and
6_ const-cast
7_ and-eq
8_ constexpr
9_ asm
10_ continue
11_ auto
12_ decltype
13_ bool
14_ default
15_ break
16_ delete
17_ case
18_ do
19_ catch
20_ double
21_ char
22_ dynamic-cast
23_ char 16-t
24_ else
24_ char 32-t
25_ enum
26_ class
27_ export
28_ extern
29_ false
30_ float
31_ Private
32_ for
33_ Protected
34_ friend
35_ Public

36_ goto

37_ register

38_ if

39_ reinterpret-cast

40_ inline

40_ return

42_ int

43_ short

43_ long

44_ signed

45_ mutable

46_ sizeof

47_ namespace

48_ static

49_ new

50_ static-asserts

noexpect

52_ static-cast

not

54_ struct

not-eq

55_ switch

nullptr

58 template

operator

60_ or

or-eq

"Pseudocodes And Flowcharts"

Name:

Fatima Ghaffar

Roll #:

03

Department:

Computer Science

CS (2024 - 2028)

Submitted to:

Sir Waseem

Questions

1. Sum of two numbers
2. Even or odd
3. Find the Maximum
4. Factorial Calculation
5. Count to Ten
6. Temperature conversion
7. Simple interest
8. Fibonacci sequence
9. Palindrome check
10. Reverse a string
11. Length in cm
12. 10 odd Numbers
13. Average of Numbers
14. Area of Rectangle
15. Maximum of two numbers
16. Name, Grade, Average
17. Calculate 2^4
18. 2^4 using Loop
19. Num 1 and Num 2
20. Area of Circle
21. Positive or Negative Number
22. Quadratic equation
23. Sum of even numbers
24. Factorial of 20
25. Largest number
26. Summation of +ve & -ve Numbers
27. Sine of Angles
28. Square or Cube
29. Employee details
30. Pass or Fail.

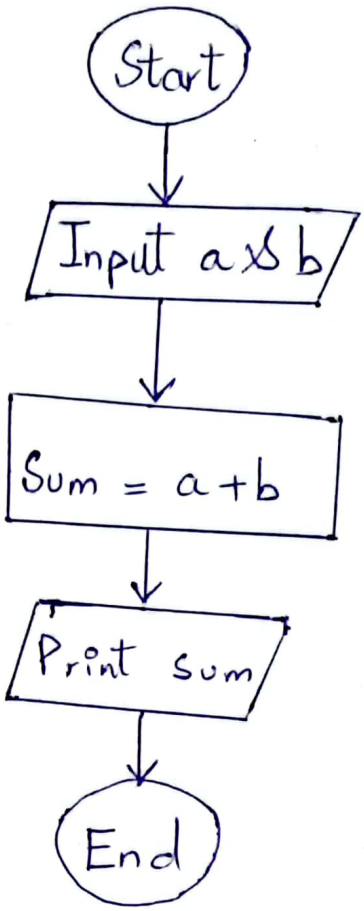
Sum of two numbers

Write a pseudocode and draw a flowchart to print sum of two numbers

•Pseudocode:

1. Start
2. Input a & b
3. Sum = a+b
4. Print sum
5. End.

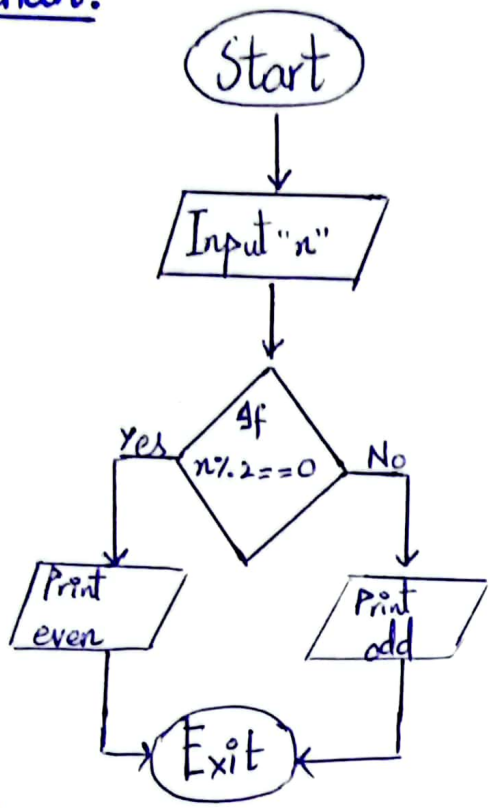
•Flowchart:



2. Even or Odd

Write a pseudocode and flowchart to determine if a given number is even or odd

Flowchart:



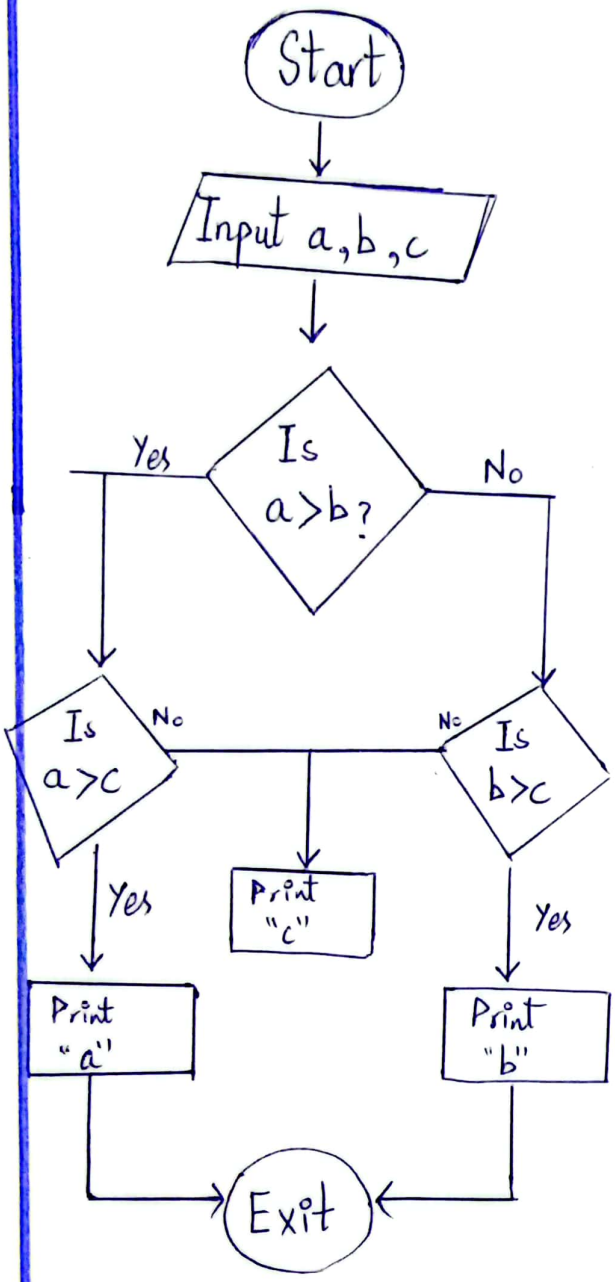
Pseudocode:

1. Start
2. Input "n"
3. If n%.2 ==0
   Print even
   Else
   Print odd
4. Exit

3 Find the Maximum

Write a pseudocode to determine if a given number is maximum.

Flowchart:



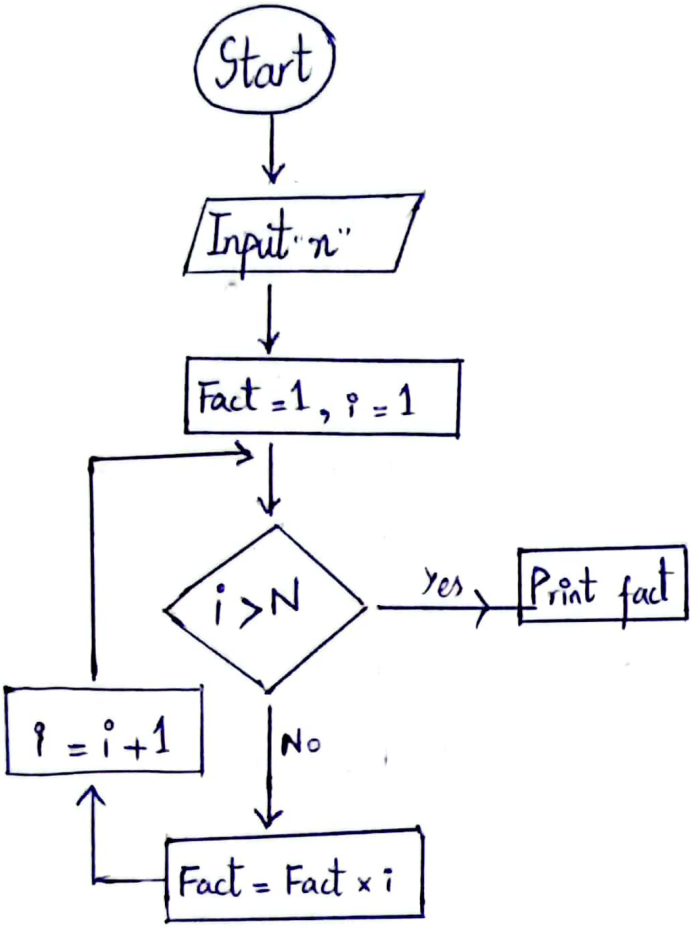
Pseudocode:

1. Start
2. Read a, b, c
3. If a>b do
   If a>c do
   Print "a"
   Else
   Print "c"
   Else
   If b>c do
   Print "b"
   Else
   Print "c"
4. Exit

4 Factorial Calculation

Write a pseudocode & flow chart to calculate the factorial of a given non-negative integer.

Flowchart:



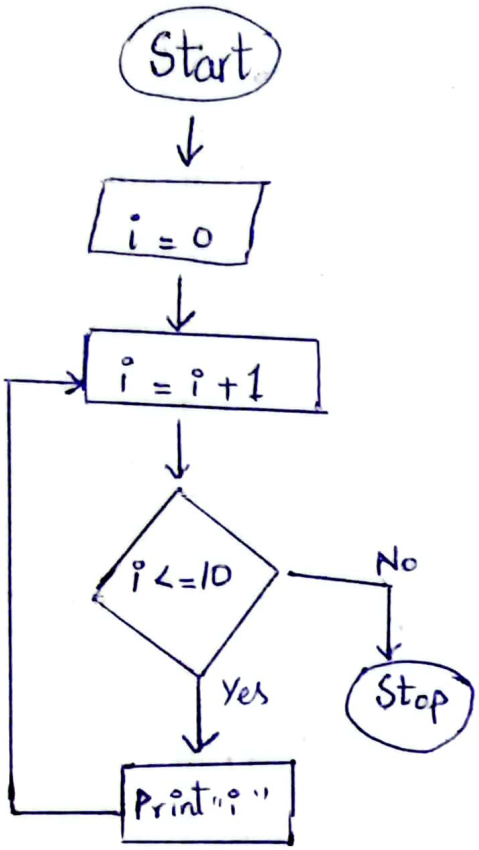
Pseudocode:

1. Start
2. Input "n"
3. fact = 1, i = 1
4. Check condition
   i <= n
5. If false,
   display fact
   If true,
   fact = fact * i
   i = i + 1
6. Print fact
7. Exit

5 Count to Ten

Write a pseudocode and flow chart that prints numbers from 1 to 10

Flow chart:



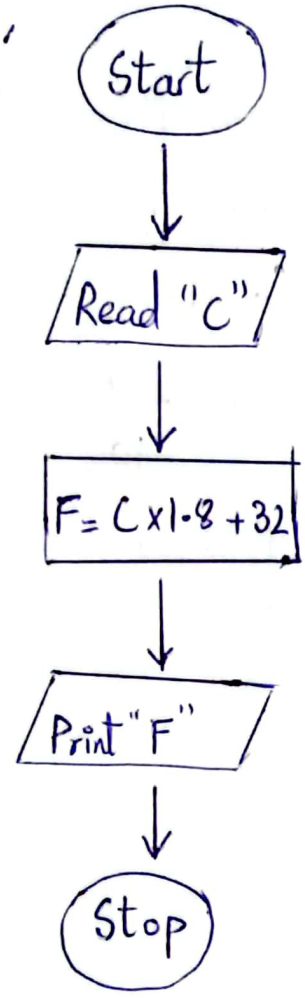
Pseudocode

1. Start
2. i = 1
3. WHILE,
   i <= 10 THEN
   Print "i"
   INCREMENT counter by 1,
   i = i + 1
   END WHILE
4. END

6. Temperature Conversion

Write a pseudocode & flowchart to convert a temperature from celcius to Fahrenhiet

Flowchart



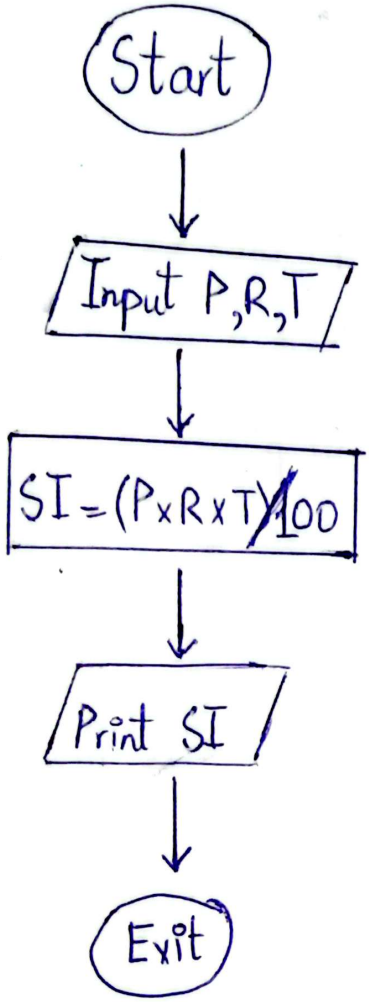
Pseudocode

1. Start
2. Read "C"
3. F = C x 1.8 + 32
4. Print "F"
5. Stop.

7. Simple Interest

Write a pseudocode and flowchart to calculate the simple interest given principale, rate and time

Flowchart



Pseudocode

1. Start
2. Input Principal(P), Rate(R), Time(T)
3. Calculate,

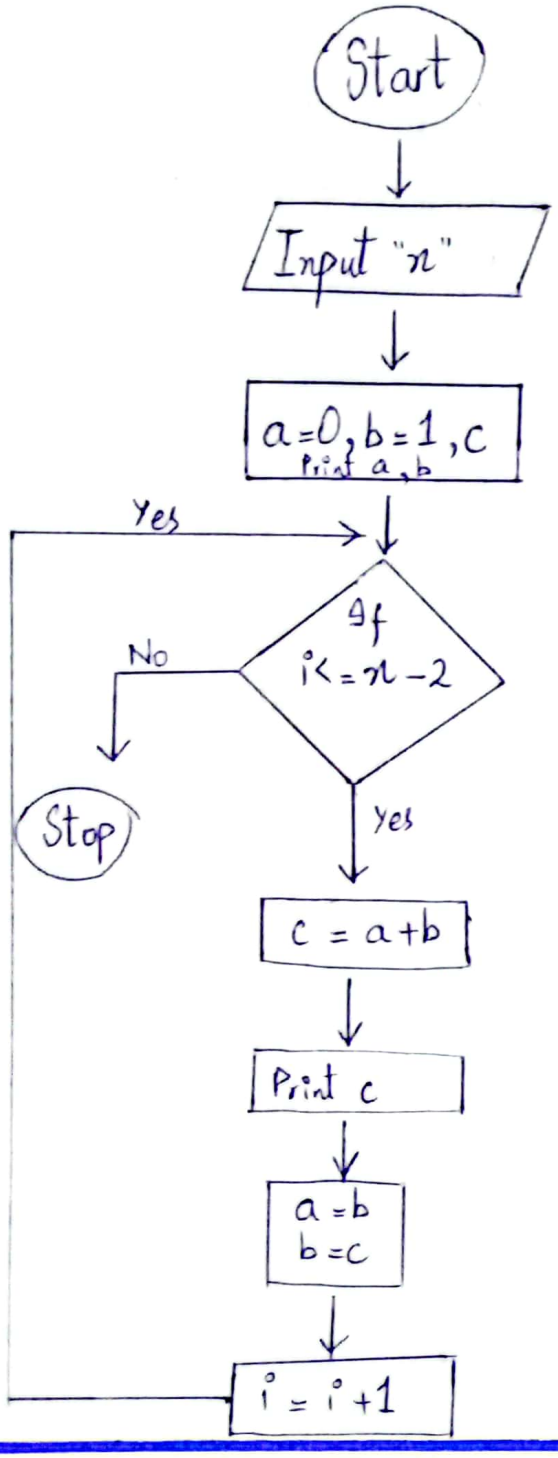
$$SI = \frac{P \times R \times T}{100}$$

4. Print SI
5. Exit

Fibonacci Series

Write a pseudocode & flowchart to generate the first n numbers in the Fibonacci series

Flow chart:



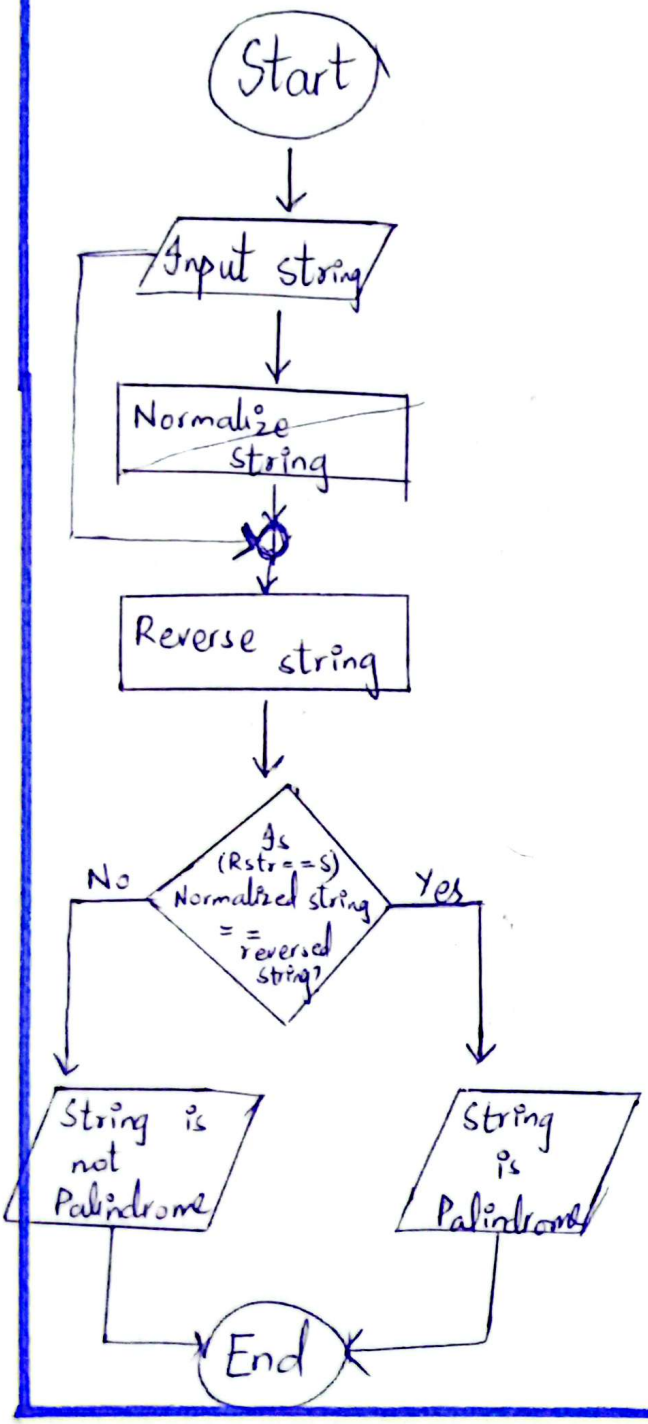
Pseudocode:

1. Start
2. Input "n"
3. Declare n, a, b, next, i
4. Print "Enter the number of terms"
5. a = 0
   b = 1
6. Print "Fibonacci sequence"
7. FOR i FROM 1 to N Do
   Print a
   next = a+b
   a = b
   b = next
   END FOR
8. END

9. Palindrome Check

Write a pseudocode and flowchart to check if a given string is Palindrome

Flowchart:



Pseudocode:

1. Start
2. Input string
3. Normalize string
4. Rerverse string
5. Compare strings,
6. Is normalized string == Reversed string

   Yes

   The string is palindrome

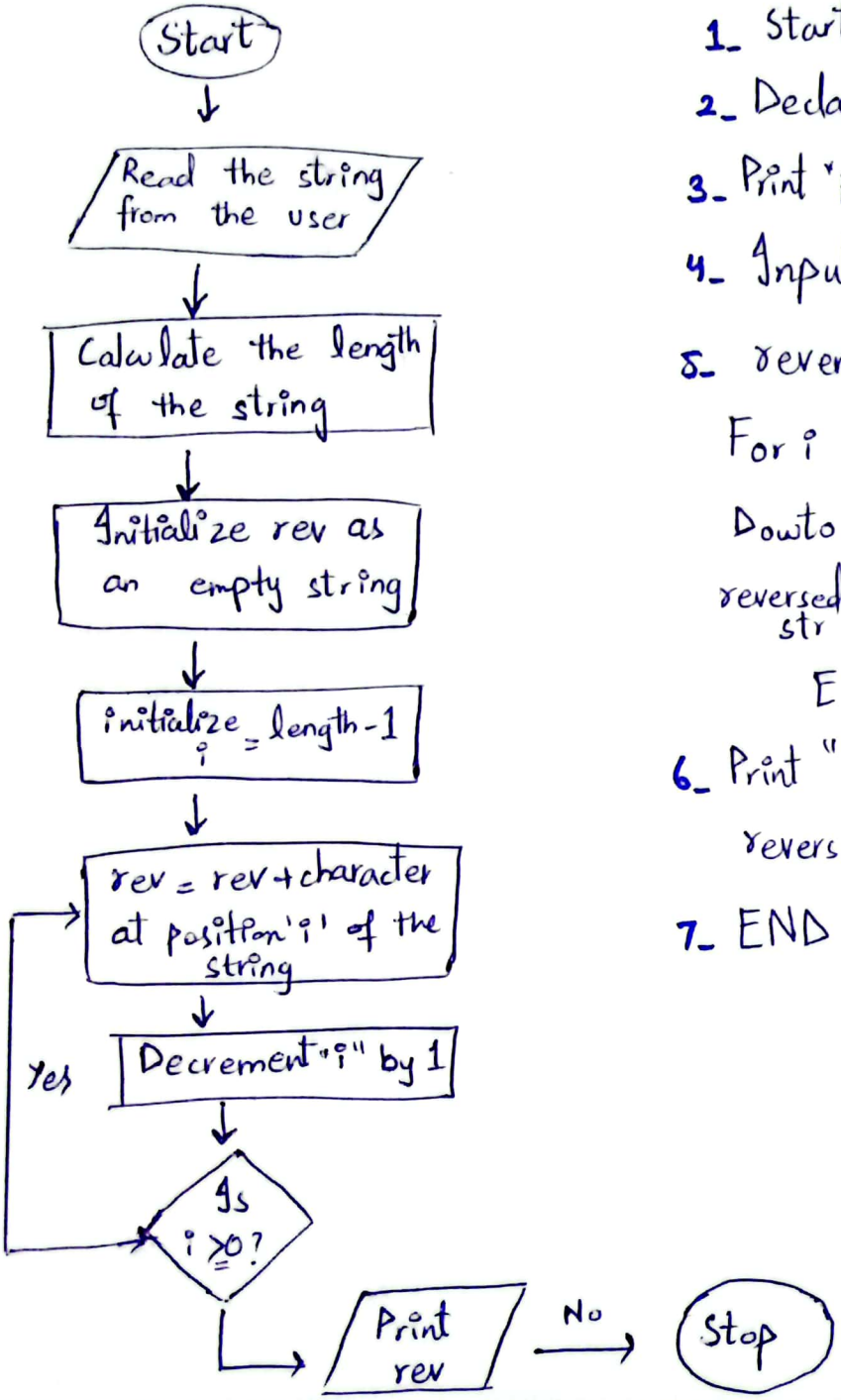
   No

   The string is not Palindrome
7. End

10 Reverse a string

Write a pseudocode & flowchart to reverse a give string and print the result.

Flowchart:



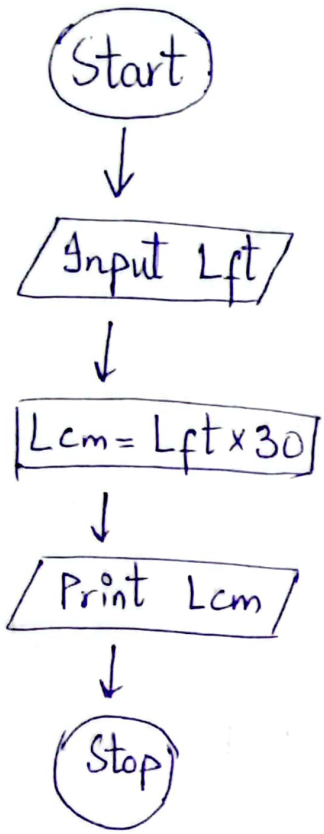
Pseudocode:

1. Start
2. Declare str, reversed str, i
3. Print "Enter a string"
4. Input str
5. reversed Str = " "
   For i from length(str)
   Downto 1 Do
   reversed str = reversed str + CHARAT (str, i)
   END FOR
6. Print " The reversed string is ";
   reversed str
7. END

Length in cm

Draw a flowchart to convert the length in feet to cm.

Flowchart:



Pseudocode:

1. Start
2. Input Lft
3. Lcm = Lft x 30
4. Print Lcm
5. Stop

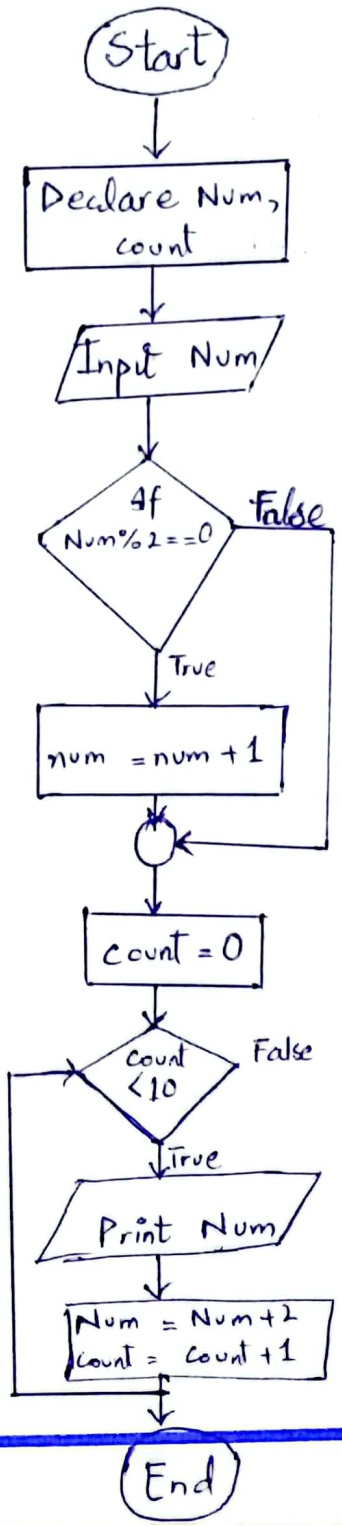
10 Odd Numbers

Write pseudocode and draw a flowchart that will print 10 odd numbers from the user.

Pseudocode

1. Start
2. Declare, count, Num
3. Input Num
4. If num % 2 == 0, then
   num = num + 1
   count = 0
   Else,
   count = 0
   If count < 10,
   Print Num
   Num = Num + 2
   Count = count + 1
   Else
5. End.

Flowchart



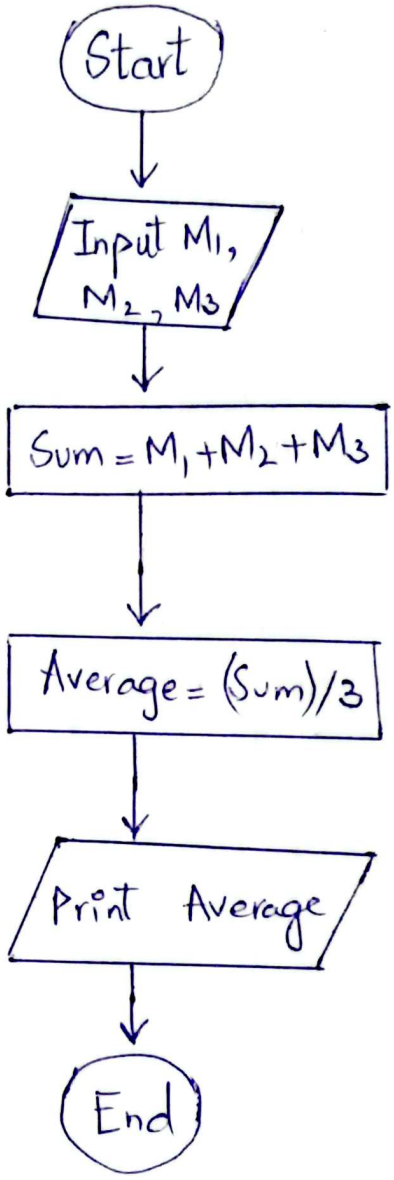
Average of Numbers

Write a pseudocode and flowchart to find the average of 3 numbers

Pseudcode:

1. Start
2. Input M_1, M_2, M_3
3. Sum = $M_1 + M_2 + M_3$
4. Average = Sum / 3
5. Print sum and average
6. End

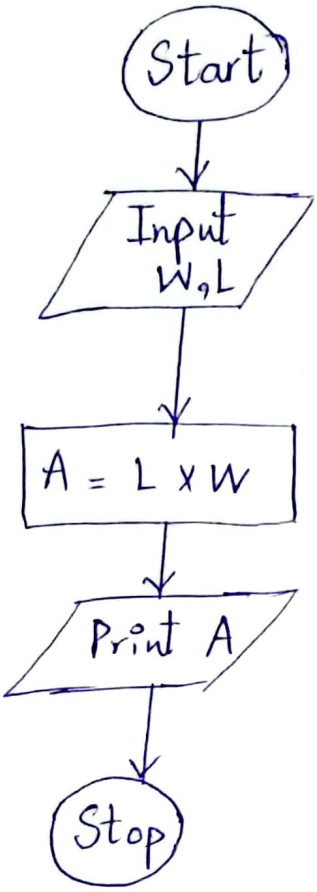
Flowchart:



Area of Rectangle

Draw a flowchart to read the two sides of rectangle and calculate its area.

Flowchart:



Pseudocode:

1- Start

2- Input L,W

3- A = L x W

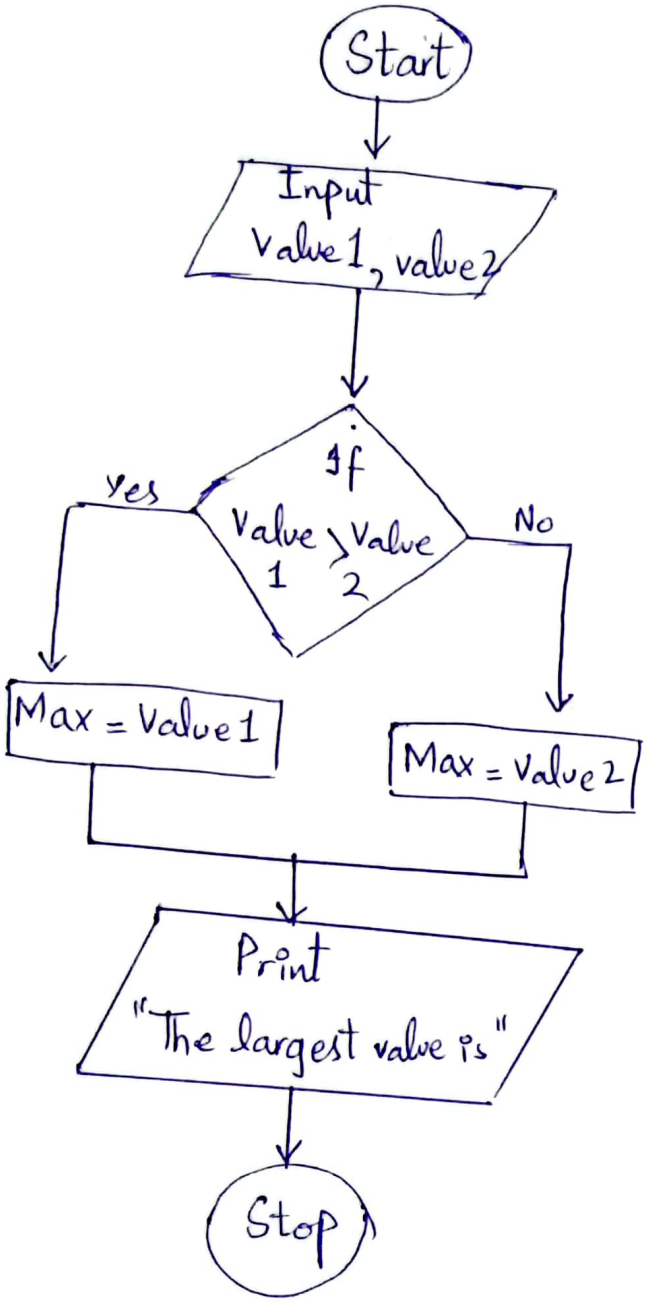
4- Print A

5- Stop

Maximum of 2 numbers

Draw a flowchart to read two values, determines the largest value with an identifying message

Flowchart

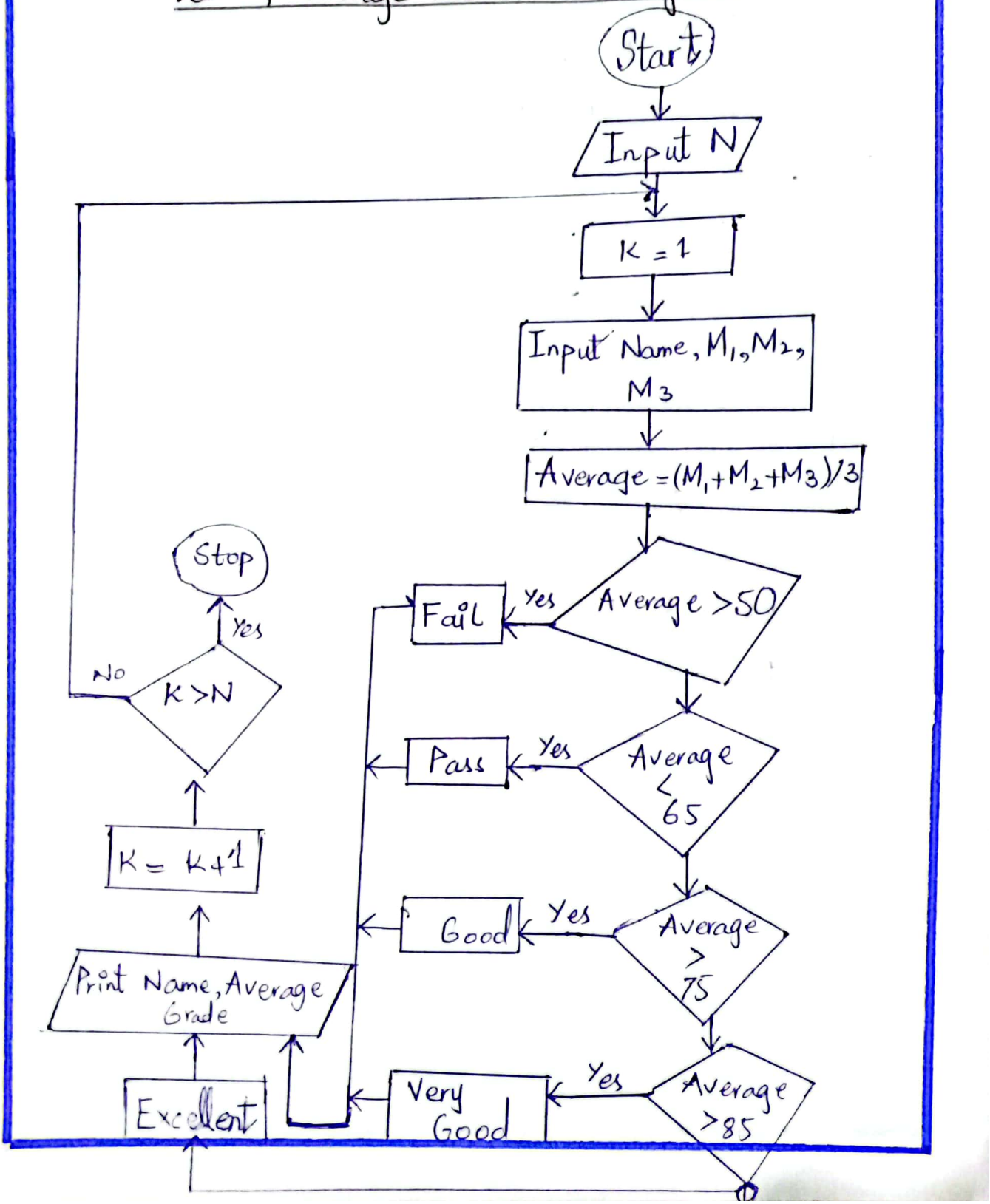


Pseudocode

1. Start
2. Real value 1 and value 2.
3. If value 1 > value 2
   Print value 1
   Else
   Print value 2
4. Exit

Name, Grade, Average

Draw a flowchart to enter the name of Nstudents and print the student name, average marks and grade.



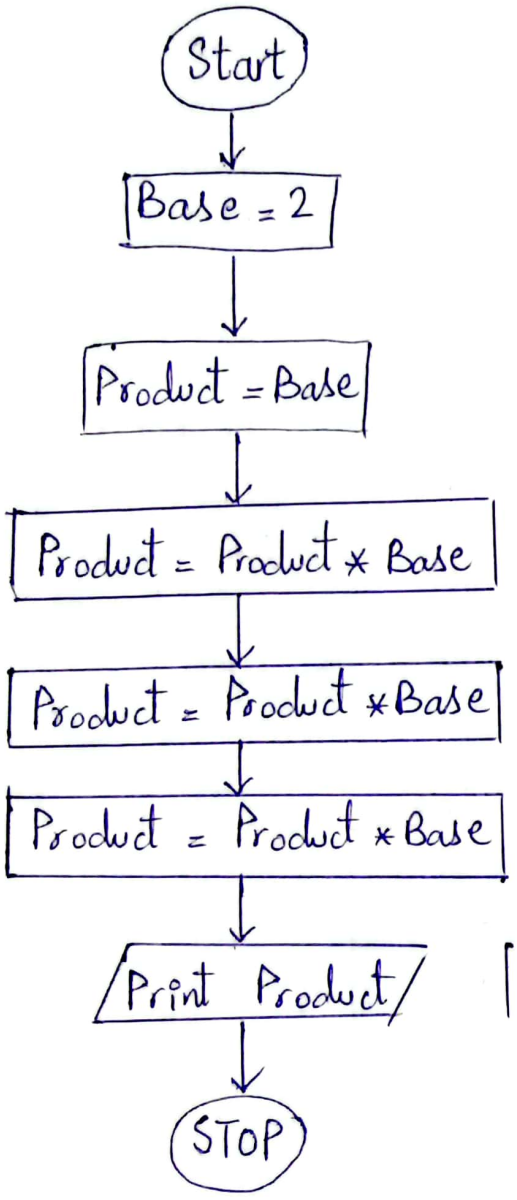
Pseudocode:

1_Start
2_Input N
3_K = 1
4_Input name M_1, M_2, M_3
5_Averge = $(M_1 + M_2 + M_3)/3$
6_Check, Average > 50
    If Yes, fail
    Else, Average > 65
    If Yes, pass
    Else, Average > 75
    If yes, Good
    Else, Averge > 85
    If yes, very good
    Else, Exellent
7_Print Name, Average, grade
8_ K = K + 1
9_Check Is K > N
    If No, go to step 3
    If Yes,
10_STOP

Calculate "2^4"

Write an algorithm and draw a flowchart to calculate "2^4".

Flowchart



Pseudocode

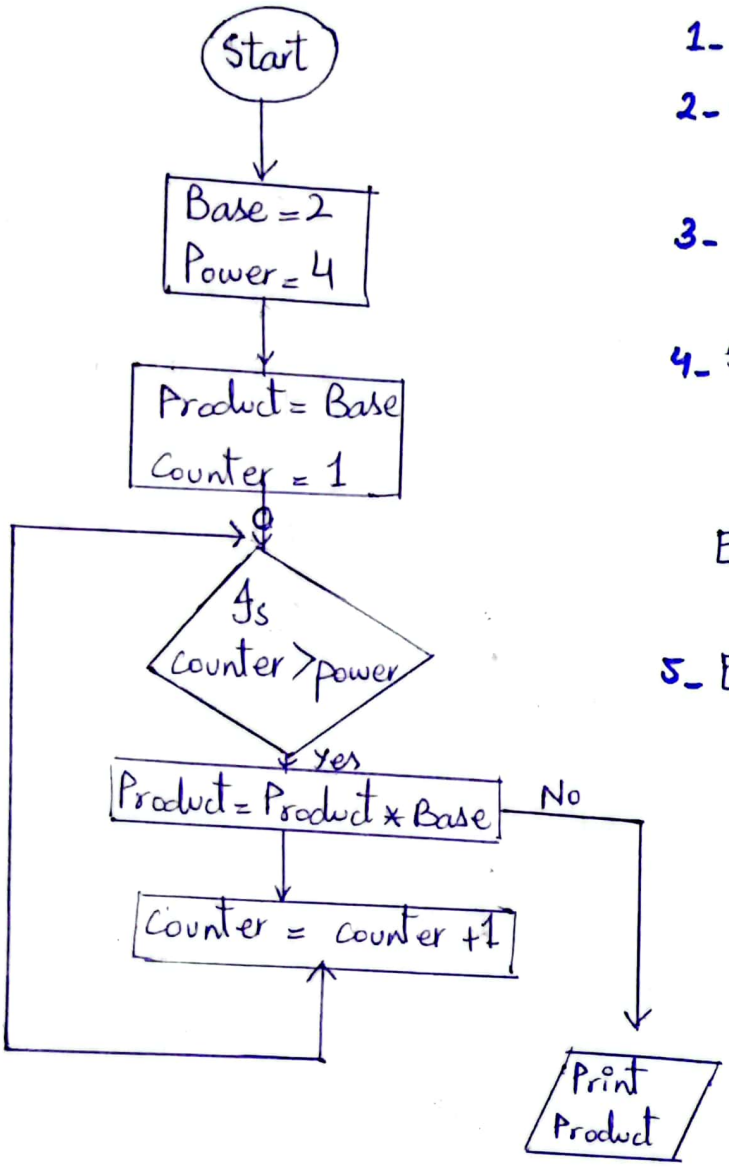
1. Start
2. Base = 2
3. Product = Base
4. Product = Product * Base

   Repeat till the power equals to "24"
5. Print Product
6. Exit

"2^4" using Loop

Write an algorithm and draw flowchart to calculate 2^4 using loop approach.

Flowchart:



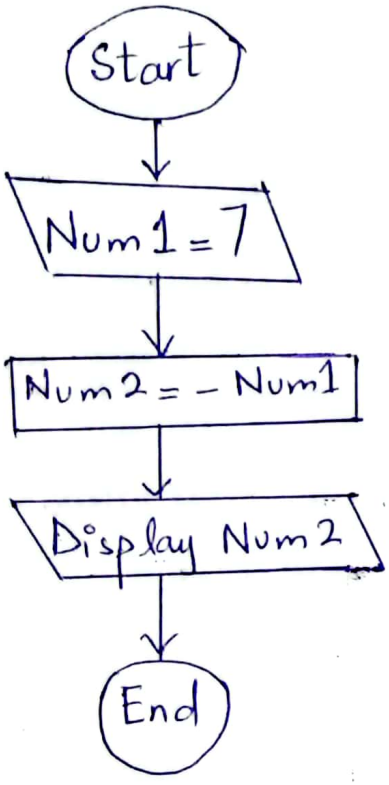
Pseudocode:

1. Start
2. Base = 2
   Power = 4
3. Product = Base
   Counter = 1
4. If Counter < Power
   Product = Product * Base
   Counter = counter + 1
   Else
   Print product
5. END

Num1 and Num2

Draw a flowchart to read a variable num1 = ₹ and store the negative value in num2

Flowchart:



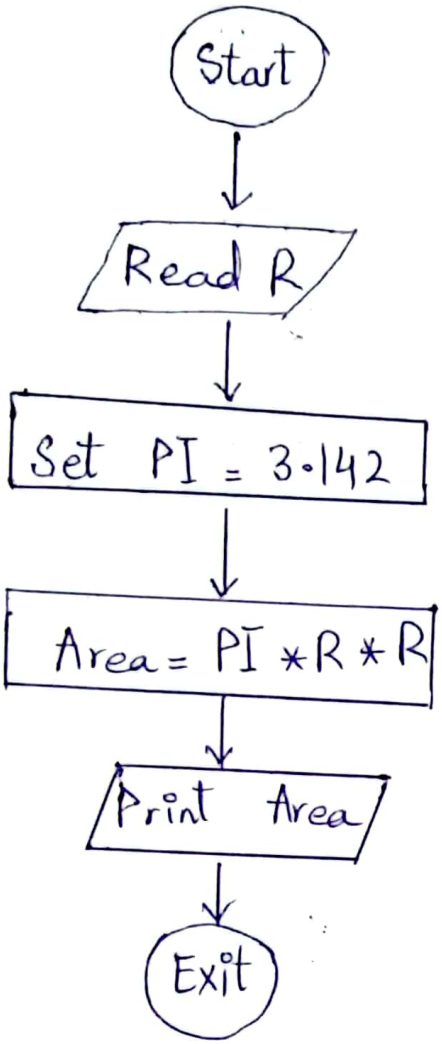
Pseudocode:

1. Start
2. Input Num1 = ₹
3. Num2 = - Num1
4. Display Num2
5. Exit

Area of Circle

Draw a flowchart and algorithm to compute the area of circle of radius "R"

Flowchart:



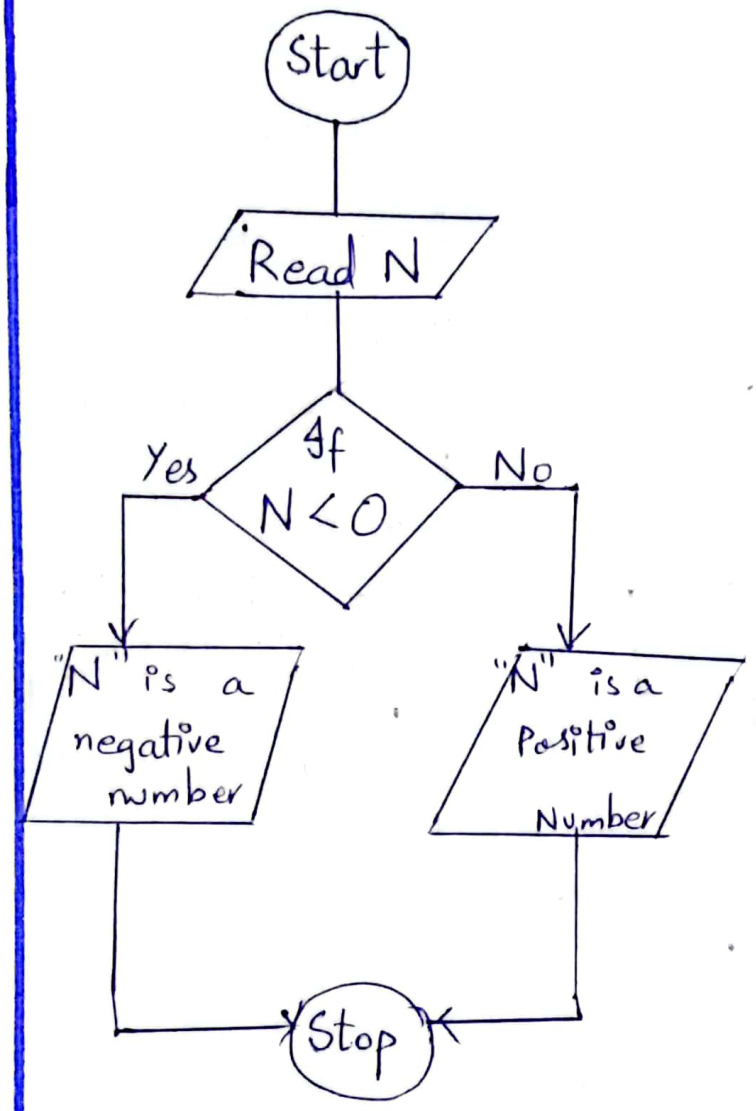
Pseudocode:

1. Start
2. Read R
3. Set PI = 3.142
4. Area = PI * R * R
5. Print Area
6. Exit

Positive or Negative Number

Draw a flowchart to read a number and intentify either the number is positive or negative

Flowchart:



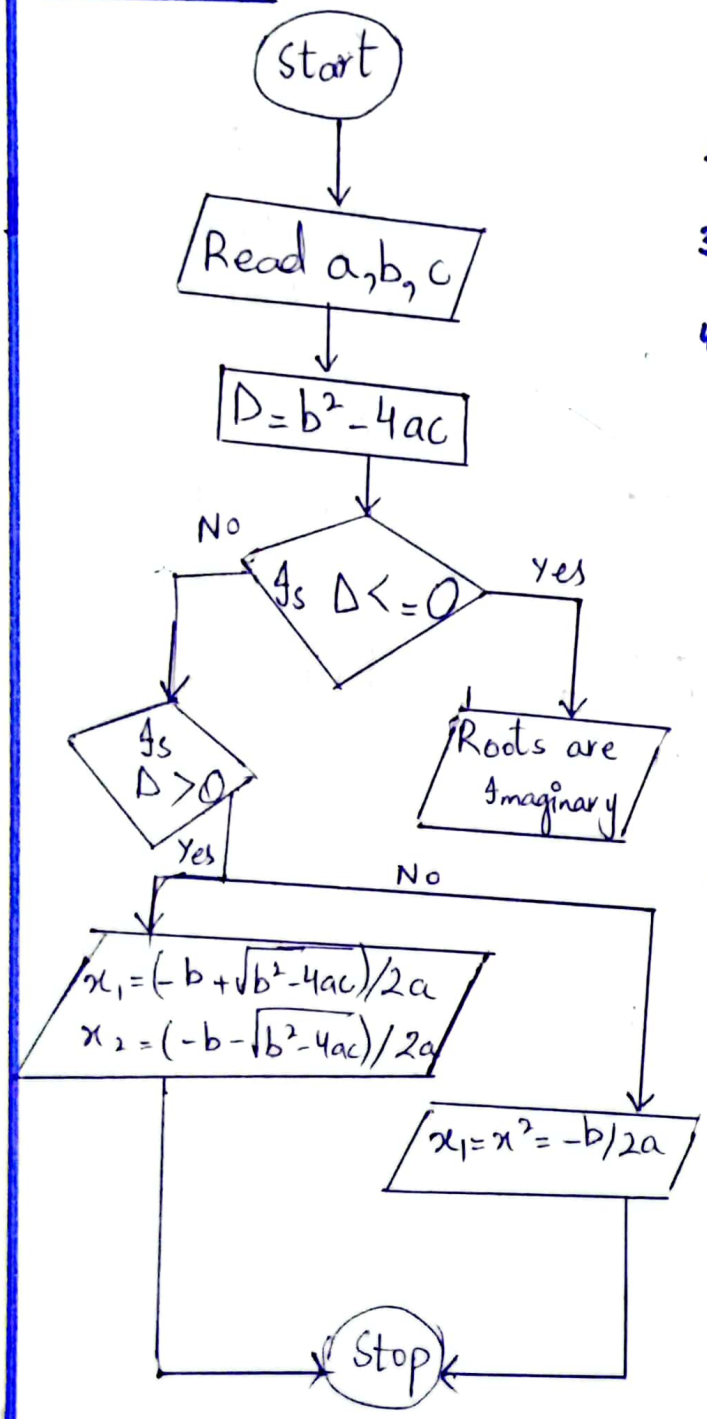
Pseudocode:

1. Start
2. Read N
3. If N<0
   Print "N is a negative number"
   Else
   Print "N is a positive number"
4. END

Quadratic Equation

Draw a flowchart and algorithm to find all the roots of quadratic equation $ax^2+bx+c=0$

Flowchart



Pseudocode

1. Start
2. Read a, b, c
3. $D = b^2 - 4ac$
4. If $D <= 0$

   Roots are imaginary

   Else,

   If $D > 0$

   $x_1 = (-b + \sqrt{D})/2a$

   $x_2 = (-b - \sqrt{D})/2a$

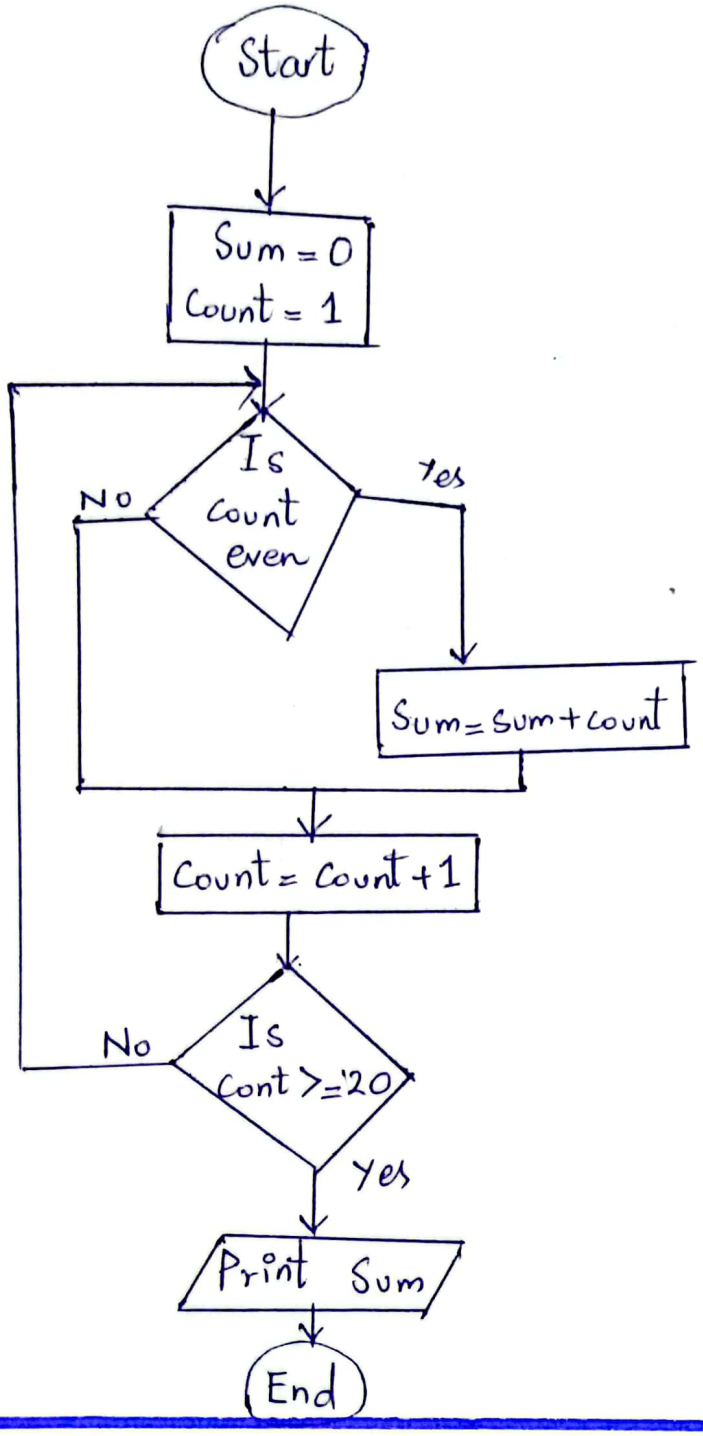
   Else,

   $x_1 = x^2 = -b/2a$
6. Exit

Sum of Even No.

Draw a flow chart to add even numbers from 0 to 20

Flowchart:



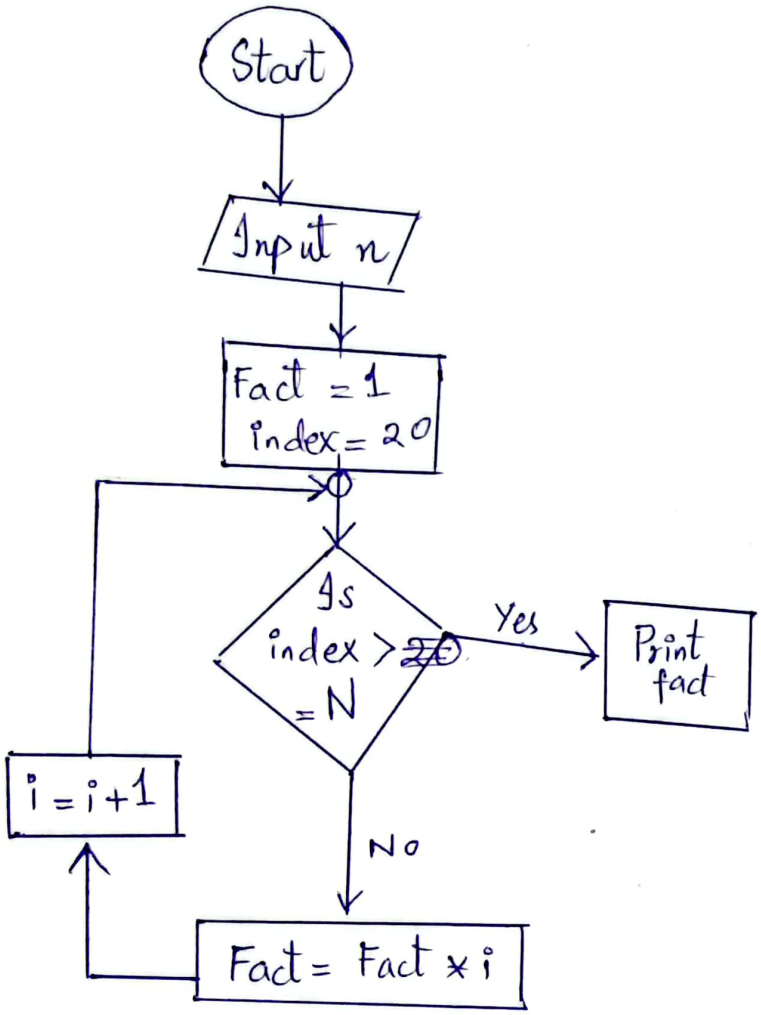
Pseudocode:

1. Start
2. Sum = 0
   Count = 1
3. Check if count even,
   Sum = sum + count
   Count = count + 1
   Check if
   Count >= 20
   If true,
   Print sum
   Else,
   go back to condition box
4. END

Factorial of 20

Draw a flowchart to obtain the factorial of 20.

Flowchart:



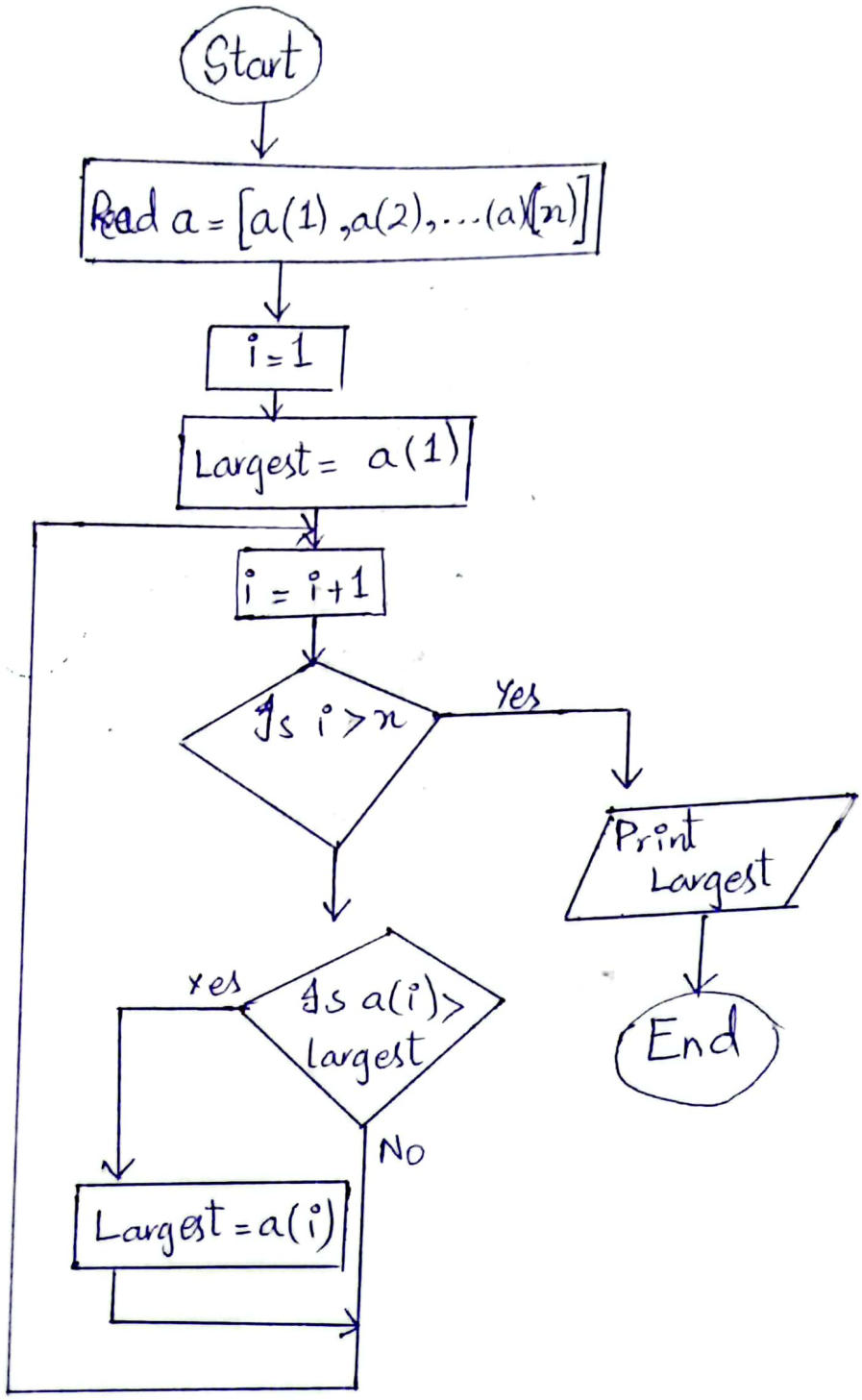
Pseudocode:

1- Start
2- Input n
3- fact = 1, i = 20
4- Check condition,
   i <= n
   If false,
      display fact
   If true,
   fact = fact * i
   i = i + 1
5- Print fact
6- Exit

"Largest Number"

Draw a flowchart find the largest number in a list of numbers

Flowchart:



Pseudocode:

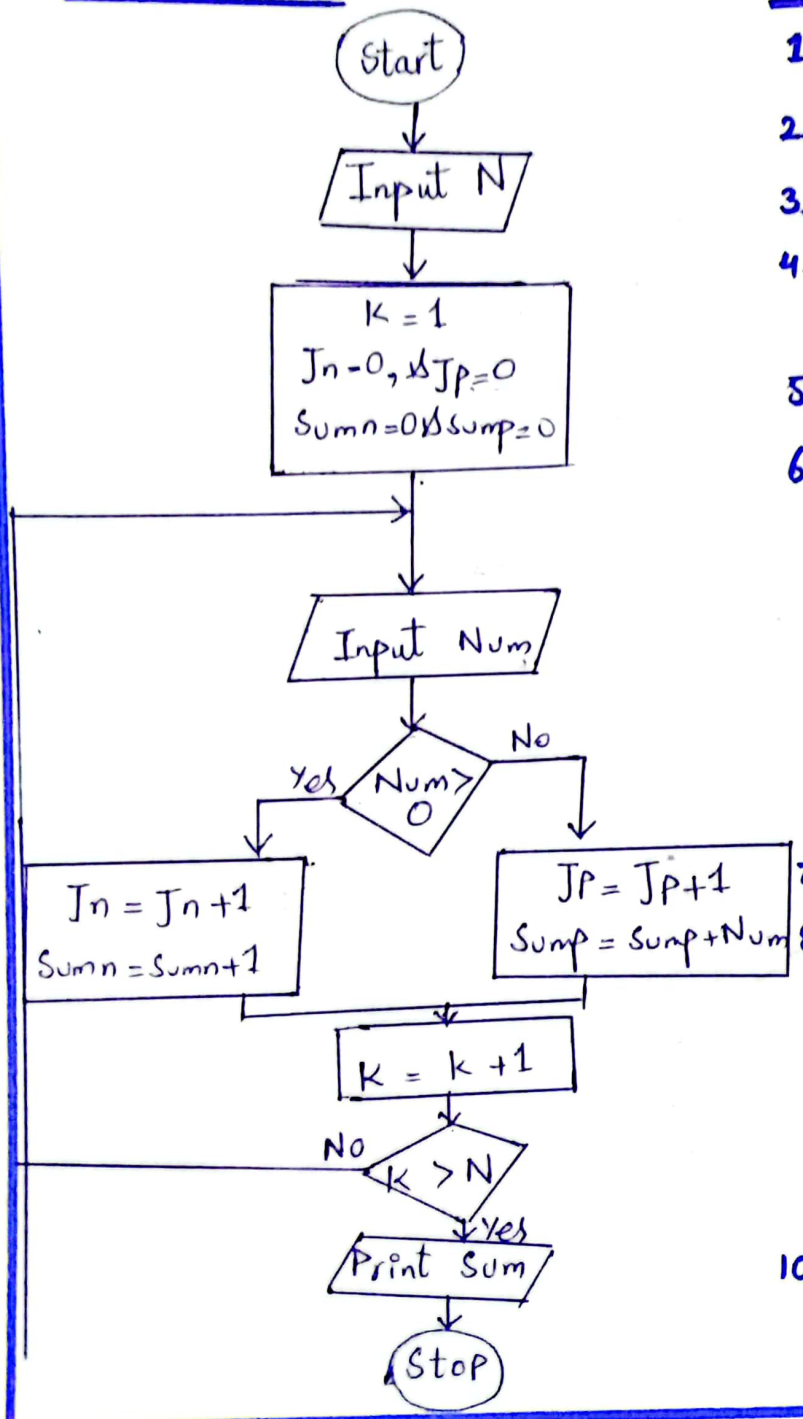
Psevdocode:

1. Start
2. Read $a = [a(1), a(2), \ldots\ldots a(n)]$
3. $i = 1$
4. Largest = $a(1)$
5. $i = i + 1$
6. Check if $i > n$
   - If true,
     - Print Largest
   - If false,
     - Check $a(i) >$ Largest
       - If false,
         - go back to step 5
       - If true,
         - Largest = $a(i)$
7. End.

Summation of +ve & -ve No.

Draw a flowchart to accept N numbers and get the summation of negative, the summation of positive numbers and the number in each group

Flowchart:



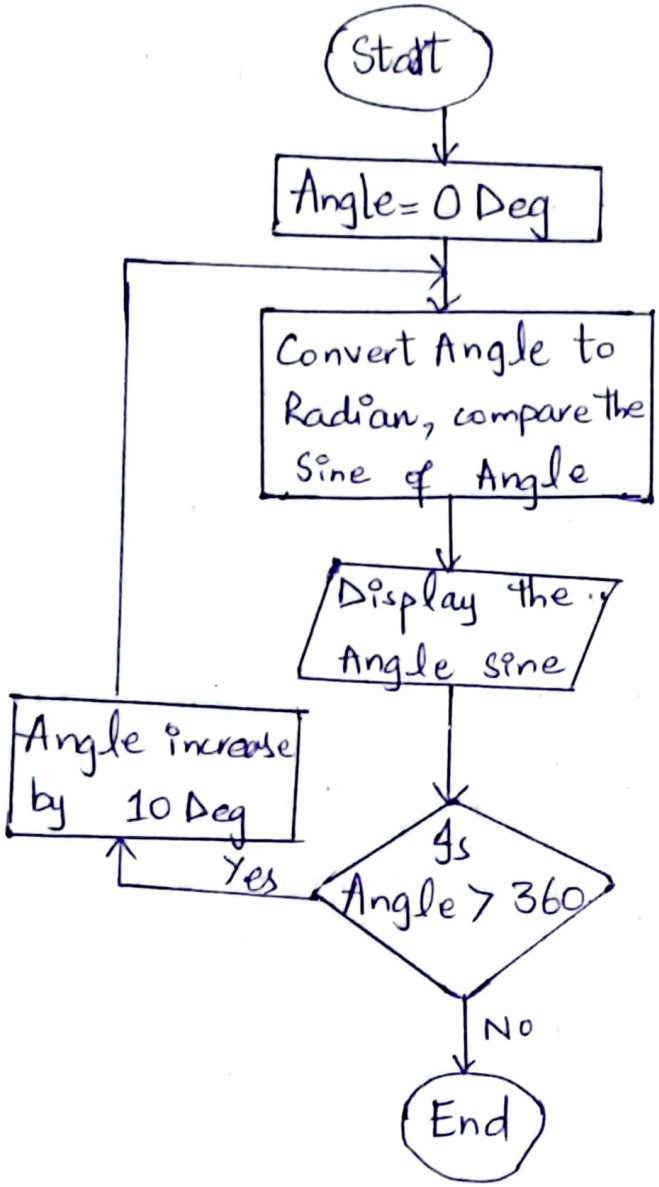
Pseudocode

1. Start
2. Input N
3. K = 1
4. Jn = 0, Jp = 0
   Sumn = 0, Sump = 0
5. Input Num
6. Is Nm > 0
   If true,
   Jp = Jp + 1
   Sump = Sump + 1
   If false,
   Jn = Jn + 1
   Sumn = Sumn + 1
7. K = K + 1
8. Is K > N
   If false,
   Go to step 4
   If true
9. Print sum
10. Exit

Sine of Angles

Draw a flowchart to obtain the sine of angles from 0 to 360° with a step of 10°

Flowchart



Pseudocode

1. Start
2. Angle = 0°
3. Convert angle to radian
4. compare the sine of angle
5. Display the angle sine
6. Is Angle > 360
   Yes, increase angle by 10°
   No,
7. END

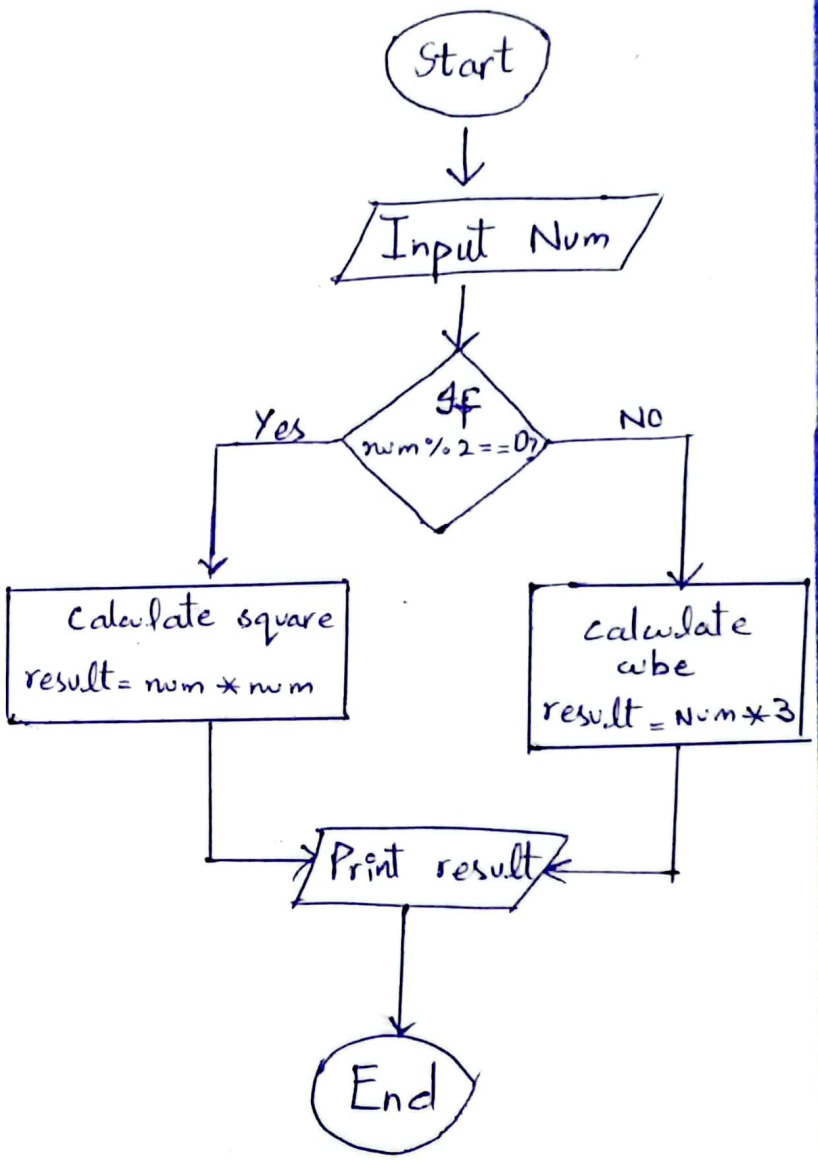
Square OR Cube

Write pseudocode and flowchart for a program that will get one number from user, print its square if it is even, Print its cube if it is odd.

• Pseudocode

1. Start
2. Input num
3. If num % 2 == 0,
   Calculate square
   result = num * num
   Else,
   calculate cube
   result = Num * 3
4. Print result
5. End.

• Flowchart



Employee Details

Write a pseudocode and draw a flowchart to read an employee NO., overtime, hours worked, hours absent & Bonus payment for 10 employee 1 by 1.

• Pseudocode:

1- Start

2- Enter Employee details
(EMPNO, NAME, ABSENT, OVERTime)

3- For i = 1 to 10

4- Effective overtime = overtime - (2/3) * ABSENT

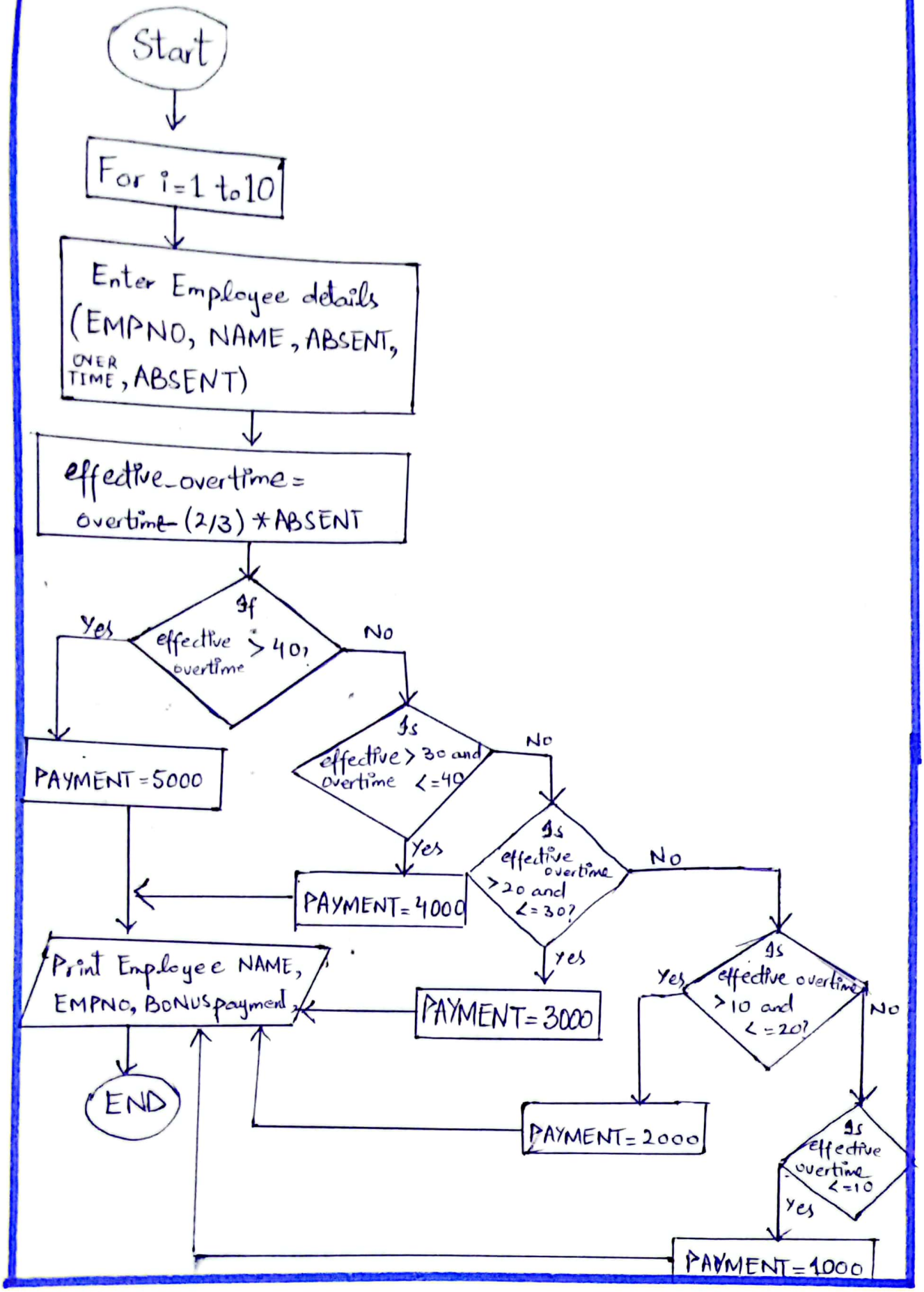
5- If effective overtime %0
Payment = 5000
Else If effective overtime > 30 and < = 40,
Payment = 4000
Else If effective overtime > 20 and < = 30,
Payment = 3000
Else If effective overtime > 10 and < = 20
Payment = 2000
Else If effective overtime < = 10
Payment = 1000
END If

6- Print "EMPNO, NAME, Bonus payment"
END FOR

7- END

• Flowchart:

• Flow-chart:


Start
For i=1 to 10
Enter Employee details (EMPNO, NAME, ABSENT, OVER TIME, ABSENT)
effective_overtime = overtime - (2/3) * ABSENT
If effective overtime > 40?
Yes
No
PAYMENT = 5000
Is effective > 30 and overtime <=40
Yes
No
PAYMENT = 4000
Is effective overtime >20 and <=30?
Yes
No
PAYMENT = 3000
Is effective overtime >10 and <=20?
Yes
No
PAYMENT = 2000
Is effective overtime <=10
Yes
PAYMENT = 1000
Print Employee NAME, EMPNO, BONUS payment
END

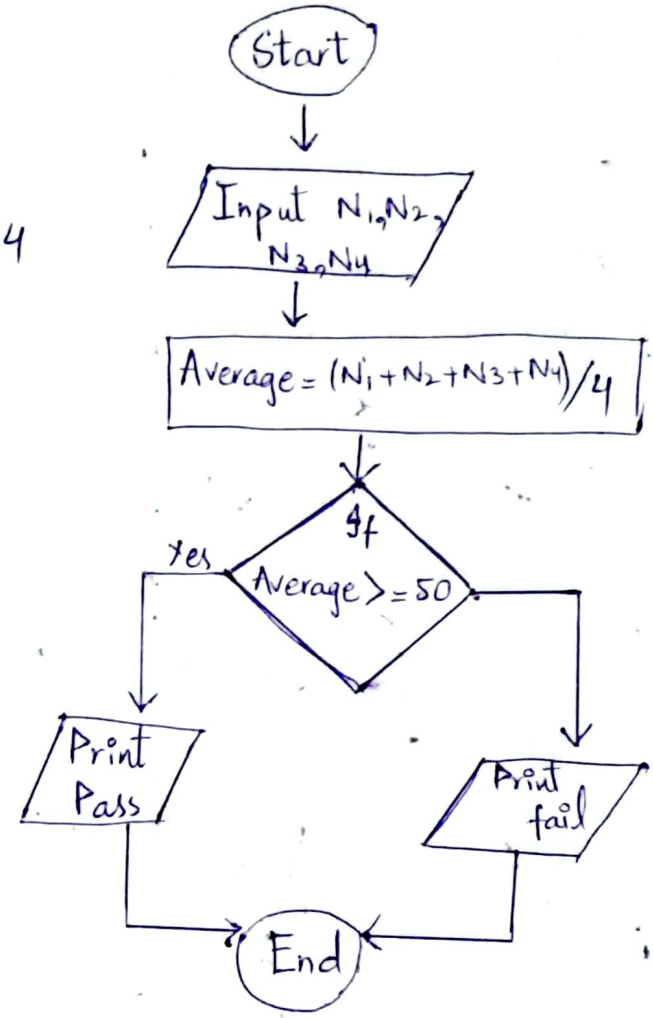
Pass or Fail

Write a pseudocode and flowchart to find whether it is pass or fail.

• Pseudocode

1. Start
2. Input N_1, N_2, N_3, N_4
3. Average = $(N_1+N_2+N_3+N_4)/4$
4. If Average >= 50,
   Print pass
   Else,
   Print fail
   End If
5. End.

• Flowchart:



About C++

Introduction, Applications, Drawbacks,
ASCII Table, keywords, Stuff
Operator Precedence

Name:

Fatima Ghaffar

Roll #:

BSCSF24M03

Subject:

Programming Fundamentals

Department:

Computer Science
CS (2024-2028)

Submitted to:

Sir Waseem

About C++

Introduction to C++

- C++ is object oriented programming (OOP) language It was developed by "Bajarne Stroustrup" in 1980.
- Bajarane Stroustrup wants to develop a language that supports object-oriented programming features with the powerful features of C++ language, and hence C++ evolved.
- C++ consists of C language features alongwith class object construct features of Simula67.
- Initially, C++ is named as **"C with classes"** by Bjarane stroustrup but later in **1983**, it was renamed to **C++** (indicating an increment to C) Proposed by **Rrick Mascitti.**
- It was a superset of "C" language, i.e: C++ contains features of "C" language as adds some of its own. Almost all programms written in C also **C++ Programs.**
- C++ programming language provides many essential features including polymorphism, virtual and friend functions, templates, namespaces and operator over-loading e.t.c.

Applications using C++:

Game Engines

C++ is one of the best at the field of game development and many game engines including Visual Pinball (Developed game - PinBall), Frostbite (Developed game - Need for speed), REDengine (Developed game - The Witcher), Shiva (Developed game - Prince of Persia 2) and many more are developed using C++.

GUI Based Application

Most of the adobe applications including illustrator, photoshop, e.t.c are developed in C++. Windows media player is also developed using C++.

Browsers

Mozzila Firefox is developed in C++. Also Google's chrome have some part in C++.

Advance Graphics software

Aliaa system's Maya 3D software that is used for animation, 3D graphics and virtual reality, is developed in C++.

Databases

The World's most popular databases Oracle database, MySQL e.t.c are coded in C++ along with C.

Structured Programming Language

C++ is a structured programming language ie: the whole code is divided into smaller parts (modules) with the help of functions, classes and objects. Such codes are easy to understand & modify.

Mid-Level Language

C++ is a middle-level language as it has features of both high-level (development of the program is straightforward, simple and more understandable) and low-level languages (direct interaction with hardware).

Rice Library

C++ provides a lots of in-built functions. This saves time and makes development faster.

Better Memory Management

C++ provides better memory management. It supports dynamic memory allocation and we can allocate and deallocate (free) the allocated memory at any time when needed.

Exception Handling

C++ provides Exception handling mechanism to deal with the Runtime exceptions.

Operating System

Most of the Windows OS have there huge portion written in C++ (along with visual C++). Some part of Apple OS is also written in C++.

Programming Languages

Many modern programming languages including C#, D, Java, PHP, Python, Limbo, PHP e.t.c are influenced by the C++ language and have their initial and partial implementations written in C++.

Features And Advantages of C++ Language

Gaming Simple

C++ is simple to understand as the syntax is almost similar to C. language with very little changes.

Portable Language

C++ is a portable language. It means that C++ programs written for a computing environment can easily run on a computer having the same environment.

Object-Oriented

It follows like the concept of oop like polymorphism, inheritance, encapsulation, obstraction. This makes development and maintenance easier.

Features and Advantages OF C++ Language

Some drawbacks of C++:

- C++ provides data security but it is not up to the mark, hence there are security issues with C++.
- Exception handling mechanism provided by C++ is not that much reliable.
- C++ is not a strictly typed language, i.e: we can pass an integer value to a floating type variable or vice-versa.
- C++ does not provide automatic memory management mechanism.
- C++ does not provide built-in multi-threading mechanism

ASCII Table

ASCII code (in decimal)	Character / Symbol
First 32 (0 – 31) are non-printing characters	
0	NUL (nul)
1	SOH (start of heading)
2	STX (start of text)
3	ETX (End of text)
4	EOT (end of transmission)
5	ENQ (enquiry)
6	ACK (acknowledge)
7	BEL (bell)
8	BS (backspase)
9	TAB (horizontal tab)
10	LF (NL line feed, new line)
11	VT (vertical tab)
12	FF (NP from need, new page)
13	CR (carriage return)
14	SO (shift out)
15	SI (shift in)
16	DLE (data link escape)
17	DC1 (device control 1)
18	DC2 (device control 2)
19	DC3 (device control 3)
20	DC4 (device control 4)
21	NAK (negative acknowledge)
22	SYN (synchronous idle)
23	ETB (end of trans. block)
24	CAN (canced)
25	EM (end of medium)
26	SUB (subtitle)
27	ESC (escape)
28	FS (file seperator)
29	GS (Group seperator)
30	US (Unit separator)

From (32-127) all are Printing Characters

32	SPACE	66	B
33	!	67	C
34	"	68	D
35	#	69	E
36	$	70	F
37	%	71	G
38	&	72	H
39	'	73	I
40	(	74	J
41	)	75	K
42	*	76	L
43	+	77	M
44	,	78	N
45	-	79	O
46	.	80	P
47	/	81	Q
48	0	82	R
49	1	83	S
50	2	84	T
51	3	85	U
52	4	86	V
53	5	87	W
54	6	88	X
55	7	89	Y
56	8	90	Z
57	9	91	[
58	:	92	\
59	;	93	]
60	<	94	^
61	=	95	_
62	>	98	`
63	?	97	a
64	@	98	b
65	A	99	c

From (100 – 127) are Printing characters

100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	|
125	}
126	~
127	DEL

C++ Keywords

Keyword	Description
alignas	Used to specify the alignment, requirement of a type/project
alignof	Returns the alignment in terms of bytes
and	Alternative to && operator
and_eq	Alternative to && operator
asm	Used to insert an assembly instruction
auto	Used to define a local (automatic) variable
bitand	Alternative to bitwise & operator
bitor	Alternative to bitwise | operator
bool	Boolean datatype
break	Used to pass the control out of 'while', 'do', 'for', 'switch' statement.
case	Used in switch statement to define a paticular case value.
catch	Used in Exception Handling, alongwith "try" block.
char	Basic data type. Used to indicate character.
char16_t	Datatype for UTF-16 character representation.
char32_t	Datatyper fou UTF-32 character representation.
class	Used to declare a class.
compl	Alternative to ~ operator
concept	Used to declare a named set of requirements
const	Used to make variable/function unmodifiable
constexpr	Used to declare a constant expression.
const_cast	Used to cast const variables.
continue	Used to skip certain statement inside loop.
decltype	Returns the type of the expression
default	Used in switch case. Default case get exeuted when the value doesn't match any of 'case' values
delete	Used to deallocate dynamic memory.

do	do-while loop (iteration statement)
double	Data type for floating type variables (double precision)
dynamic_cast	Used to implement runtime casting
else	Decision making statement. (Used with if)
enum	Used to define enumerated type data type i.e: a set of integer constants
explicit	Used to define the constructors that don't allow implicit conversion.
export	Used to seperate template definitions fro their declarations
extern	Used to tell the compiler that the variable is already declared somewhere else
false	Represents a boolean false value.
float	Data type for floating type variables (single precision)
for	for loop (iteration statement)
friend	Used to define a friend function to a class. We can access Private members of a class via friend function
goto	Unconditional jump statement
if	Decision making statement (Used with "else" or alone
inline	Used to define an inline function
int	Basic data type. Used to indicate integer type
along	Type modifier. Used to alter the meaning of ba data type (in order to increase their range, size
mutable	Used to define a mutable (changeable) variable inside const function.
namespace	Used to define a new scope in order to prevent name conflicts
new	Used to allocate dynamic memory for a new variable
noexcept	Used to determine whether a function or operato will throw exceptions or not
not	Alternative to ! operator

not_eq	Alternative to != operator
nullptr	Pointer literal
operator	Used to define overloaded operator functions.
or	Alternative to || operator
or_eq	Alternative to |= operator
Private	Access specifier. Declare Private members of a class
Protected	Access specifier. Declare Public members of a class
register	Used to tell the compiler to store the variable in a CPU register.
reinterpret_cast	Used to change the type of a variable.
requires	Used to indicate an associated constraint
return	Used to return the flow of control back to the calling function from the called function (with a returning value).
short	Type modifier. Used to alter the meaning of base data type
signed	Type modifier. Used to alter meaning of base data type
sizeof	Special operator. Used to determine the size of any entity (variable, datatype) in bytes.
static	Indicate static storage class. Static variable stay alive (in memory) until end of program.
static_assert	Used to perform compile-time assertion checking.
static_cast	Type conversion. Used to cast a pointer to a base class to a pointer to a derived class
struct	Used to create a structure i.e: collection of variables having different datatype
switch	Decision making statement which is used to test the value of expression against the different case values
This.	A special pointer that points to the current object.

Operator Precedence in C++

Operator	Meaning of Operator	Associativity	Rank
::	Scope resolution	Left-to-Right	1
++ -- ype ( ) type { } ( ) [ ] . -> typeid onst_cast ynamic_cast einterpret_cast statpc_cast	Suffix / Postfix increment and decrement Function-style type cast Functional call (Paretheses) Array subscripting Element selection by reference Element selection through pointer Run-time type information Cast away const Run-time type-checked cast Cast one type to another Compile-time type-checked cast	Left-to-Right	2
++ -- + - ! ~ (type) * &	Prefix increment Prefix decrement Unary plus Unary minus Logical NOT Bitwise NOT C-style type cast	Right-to-Left	2

sizeof () new, new [] delete, delete[]	Indirection (dereference) Adress-of Size-of Dynamic memory allocation. Dynamic memory deallocation		
.* ->*	Pointer to member	Left-to-Right	4
* / %	Multiplication Division Remainder (Modulo-division)	Left-to-Right	5
+ -	Addition Subtraction	Left-to-Right	6
<< >>	Bitwise Left shift Bitwise Right shift	Left-to-Right	7
< <= > >=	Less than Less than or equal Greater than Greater than or equal	Left-to-Right	8
== !=	Equal to Not equal to	Left-to-Right	9
&	Bitwise AND	Left-to-Right	10
^	Bitwise XOR (exclusive OR)	Left-to-Right	11
|	Bitwise OR (inclusive OR)	Left-to-Right	12
&&	Logical AND	Left-to-Right	13
||	Logical OR	Left-to-Right	14
?:	Conditional operator	Right-to-Left	15
= *= /= ^= %= += |= -= &= <<= >>=	Assignment operator Short-Hand Assignment operators	Right-to-Left	16
throw	throw operator	Right-to-Left	17
,	comma	Left-to-Right	18