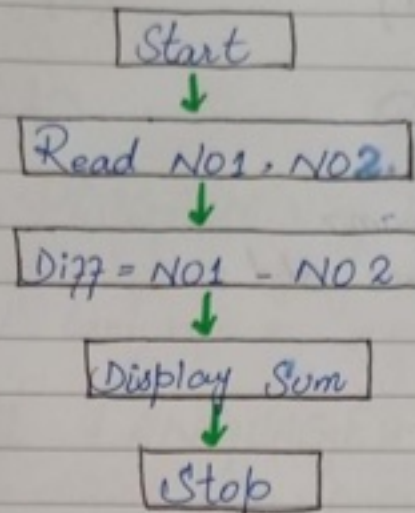


NAME. Shabir Ahmad

ID . 15226

BS (SE) .



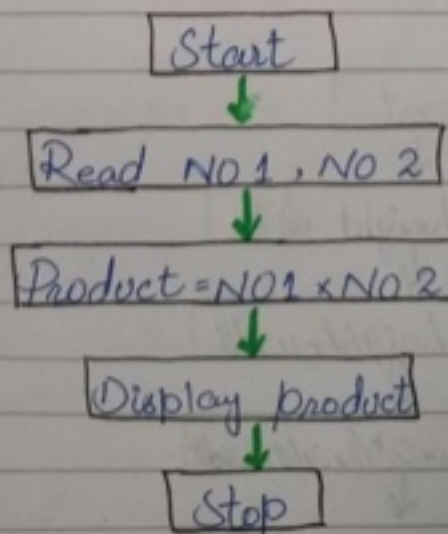
PRODUCT OF TWO NUMBERS

Input = NO 1 , NO 2

Process = multiply NO 1 & NO 2

Output = Display product.

FLOW CHART

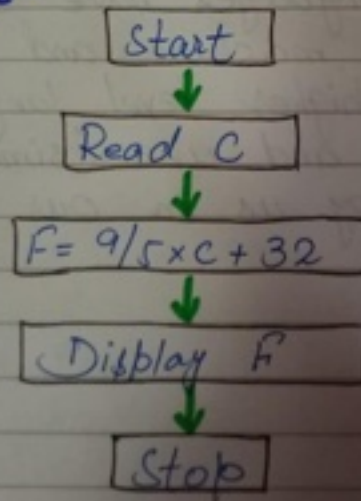


ID:15226

#03

Q # 01 (b)

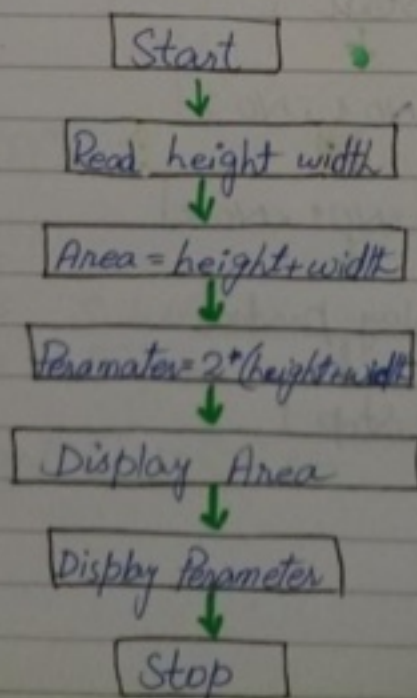
ANSWERS-



(a) Q # 02 (a)

ANSWER

```
#include <iostream>
using namespace std;
int main ()
{
    int height = 2;
    int width = 5;
    int area;
    int parameter;
    area = height * width;
    cout << "area is \n";
    cout << area;
    parameter = 2 * (height + width);
    cout << "\nparameter is \n";
    cout << parameter;
}
```



(a) Q#01. (a)

ANSWER:-

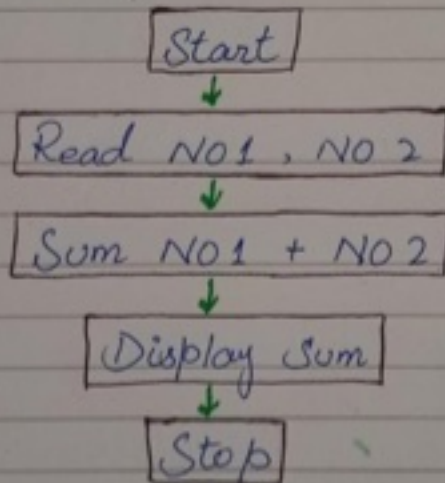
SUM OF TWO NUMBERS:-

Input = NO.1, NO.2.

Process = Add NO1 & 2.

Output = Display Sum

FLOW CHART:-



DIFFERENCE OF TWO NUMBERS:-

Input = NO 1, NO 2

Process = Subtract NO 1 & 2.

Output = Display Difference.

FLOW CHART:-

Q # 02 (b)

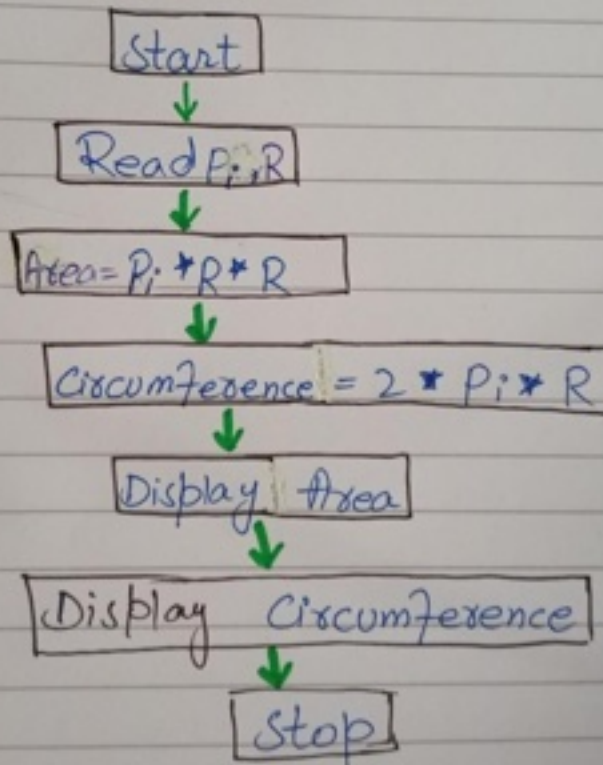
#05

ANSWER ↷

```
#include <iostream>
using namespace std;
int main ()
{
    int R = 5;
    float pi = 3.14;
    float area;
    float circumference;
    area = pi * R * R;
    cout << "Area of circle is = ";
    cout << area;
    cout << "\n circumference is = ";
    circumference = 2 * pi * R;
    cout << circumference;
}
```

ID: 15226

Date: #05



(a) Q # 03 (a)

ANSWER

TYPES OF PROGRAMING LANGUAGE

There are two types of programming language.

- * Low level language
- * High level language.

LOW LEVEL LANGUAGE

- * The next evolution in programming came with the idea of replacing binary code for instruction and addresses with symbols. Because they used symbols, these languages were first known as symbolic languages. The set of these mnemonic languages were later referred to as assembly languages.
 - * It is the first step to improve the programming structure, you should know that computer can handle numbers and letters.
 - * The set of symbols and letters from the assembly language to and a translator program is required to translate the assembly language to machine language.
 - * This translator program used for assembly language is called assembler.
 - * To program in assembly you need to understand concepts behind machine language and execution-fetch cycle of CPU.
 - * Assembly is a machine specific language.
 - * Although assembly and machine language might look similar, they are in fact two different types of languages.
- ⇒ Assembly consists of both binary and

are high level languages.
Advantages of high level languages.
Higher level languages have a major advantage over machine and assembly languages that higher level languages are easy to learn and use (similar to the languages used by us in our day to day life).

Q # 03 (b)

ANSWER-

COMPILER-

- * It is a program translator that translates the instruction of a higher level language to machine language.
- * It is called compiler because it compiles machine language instructions for every program instructions of higher level language.
- * Thus compiler is a program translator like assembler but more sophisticated.
- * A compiler can translate only those source programs, which have been written, in that language.

INTERPRETER-

- * An interpreter is another type of program translator, used for translating higher level language into machine language.

Simple words.
⇒ machine code composed only of 0's and 1's.

HIGH LEVEL LANGUAGES

- * Although assembly languages greatly improved programming efficiency, they still required programmers to concentrate on the hardware they were using. Working with symbolic languages was also very tedious, because each machine instruction had to be individually coded. The desire to improve programmer efficiency and to change the focus from the computer to the problem being solved led to the development of high level languages.
- * Assembly and machine level languages require deep knowledge of computer hardware where as in a higher language you have to know only the instructions in English words and logic of the problem.
- * Higher level languages are simple languages that used English and mathematical symbols like +, -, %, /, etc. for its program construction.
- * Any higher languages has to be converted to machine language for the computer to understand.
- * For example COBOL (Common Business Oriented Language), FORTRAN (Formula Translation) and BASIC (Beginners All-purpose Symbolic Instruction Code)

