Mid Sem

1) What is Pseudo-Code? what is the difference between Algorithm and Pseudo-Code?

Ans=> The Pseudo-Code is an informal way of writing a Program for better human understanding.

The basic difference between algorithm and Pseudocode is that an algoritm is a Step-by-step Procedure developed to Solve a Problem, while a pseudoCode is a technique of developing an algorithm.

2) What are those problems we may face without translators?

Ans=> 1) Inability to Compile Code: Translators are responsible for Converting to high-level language Code into machine Code that the Computer Can understand and execute. Without translators, we would not be able to Compile the C Code into an executable program.

2) Difficulty in detecting errors: Translators are play a Crucial role in detecting errors in the Code during the Compilation Process. Without them, we would have to manually Search for errors in the Code, which Can be time-Consuming and error-prone.

3) InCompatibility with different hardware Platforms: Translators help to ensure that the Compiled Code is Compatible with the hardware Platform on which it will run. Without them, we would have to manually write Machine Code for each different hardware Platform

Which would be impractical and error-prone.

4) Limited reusability of Code :- Translators also help in the Creation of libraries and reusable Code modules. Without them, we would have to rewrite Code for each application, Which would be time-Consuming and lead to more errors.

2) State whether the following Statement is true or false.

i) All variables must be given a type when they are declared.

Ans⇒ True : In C programming language, all variables must be given a type when they are declared.

ii) The Scanf function Can be used to read one variable at a time.

Ans⇒ False : The Scanf function Can be used to read multiple variables at a time, Separeted by Spaces, Commas, or other delimiters.

iii) In C, if data item is Zero, it is Considered as a false

Ans⇒ True : In C, if data item is zero, it is Considered as a false.

iv) The expression !(X<=Y) is Same as expression X>Y.

Ans⇒ True: The expression !(X<=Y) is Same as expression X>Y.

v) A Printf Statement Can generate only one line output.

Ans⇒ False: A Printf Statement Can generate multiple lines of output by using escape Characters Such as '\n' to indicate a new line.

vi) A Switch Statement Can always be replaced by Series of if..else statements.

Ans⇒ True.

3) a) Write a C Program using nested if..else to find out the lowest of three given numbers.

Ans⇒

```
#include <stdio.h>

int main () {
int num 1, num2, num 3;
Printf ("Enter three numbers: ");
Scan f ("%d %d %d", & num1, & num2,
                              &num3);
if (num 1 <= num 2)
{
    if (num 1 <= num3)
       Printf ("% d is the lowest number",
                              num 1);
  else
       Printf ("% d is the lowest number",
                              num 3);
   }
  else
{
    if (num2 <= num 3)
  printf ("%d is the lowest number.", num2);
else
   printf ("%d is the lowest number.", num3);
}
  return 0;
}
```

b) What will be the output of the following function if the value of "i = 5":

```
printf ("%d %d %d %d %d", i, ++i, i++, i++, ++i)
```

Ans⇒ The output of the given printf statement will be undefined and can vary depanding on the compiler and optimization level used. This is because the statement uses multiple side effects (i.e, modifying the same variable multiple times in a single expression) and the order of evaluation of function arguments is not defined by the C standard.

In this specific case, the statement has undefined behaviour because it modifies the value of 'i' multiple times in the same statement without any sequence points. The use of multiple side effects like this is not recommended in C programming because it can lead to unexpected and inconsistent behavior.

c) With an initial value $a=10$, what is the value of a if we execute $a = ++a + a++$. Explain (you ar
to consider GCC compiler).

Ans⇒ The behaviour of the expression '$a=++a + a+$
is undefined. This means that the result of this expression is not guarantee and can vary depending on the compiler and platform used.

Let's consider the behavior of the expression in GCC compiler. The order of evaluation of the expression is not guaranteed, so it could be evaluated in different orders. However, there are some rules that we can use to predict the behavior.

4) a) Write a C program to find out the largest amongst 6 (Six) given integers using ternary (?) Operator and While loop. (Do not use if else or else if Statements).

Ans ⇒

```
#include <stdio.h>

int main() {
    int a, b, c, d, e, f, largest;
    Printf ("Enter Six integers: ");
    Scanf ("%d %d %d %d %d %d", &a,
                &b, &c, &d, &e, &f);

    largest = a > b ? a : b;
    largest = c > largest ? c : largest;
    largest = d > largest ? d : largest;
    largest = e > largest ? e : largest;
    largest = f > largest ? f : largest;

    Printf ("The largest number is : %d\n",
                                largest);
    return 0;
}
```

b) Write a C Program to find out whether a give number is odd or even using Switch Statement.

Ans⇒

```
#include <stdio.h>
int main () {
    int num;

    printf("Enter an integer: ");
    Scanf ("%d", &num);

  Switch (num % 2) {
      Case 0;
        Printf ("%d is even.\n", num);
        break;
  Case 1;
        Printf ("%d is odd.\n", num);
        break;
  }
  return 0;
}
```

5. What is initialization? Why is it important? Describe the characteristics and purpose of escape sequence.

Ans⇒ Initialization is the process of assigning an initial value to a variable or a data structure when it is declared. In C programming language, initialization can be done at the time of declaration or later in the program using an assignment statement.

Initialization is important because it ensures that a variable or data structure has a valid initial value before it is used in the program. This can prevent unexpected behaviour and errors that may occur if the variable or data structure is not initialized. Initialization also makes the code easier to read and understand by providing a clear indication of the initial state of a variable or data structure.

Escape sequences are special sequences of characters that are used to represent non-printable or special characters in C programming language. Escape sequences start with a backslash '\' and are followed by one or more characters that represent the desired character or action.

The Characteristics of escape Sequences are :-

1) They always Start with a Backslash '\'.

2) They are followed by one or more Characters that represent the desired Character or action.

3) They are used to represent non-printable or Special Characters in C Programming language.

4) They Can be used in Character literals, String literals, and printf() and Scanf() format Strings.

—— x —— x —— x ——