CHAPTER NO 3: - (CONDITIONAL LOGIC)

SHORTS QUESTIONS/ANSWERS

1. ***Define controlling statement?***
   Controlling statements are programming constructs that control the flow of execution of statements in a program based on certain conditions or loops. They determine the order in which instructions are executed. There are three types of controlling statement are as follow: -
   1. *Sequential control statement*
   2. *Selection control statement*
   3. *Repetition control statement*
2. ***How many types of controlling statement?***
   There are three types of controlling statement are as follow: -
   1. *Sequential control statement*
   2. *Selection control statement*
   3. *Repetition control statement*
3. ***Define sequential control statement?***
   A sequential control statement refers to the default mode of execution in a program where statements are executed one after the other in the exact order they appear. Each statement runs exactly once, ensuring a linear and predictable flow of execution
4. ***What is the use of if statement?***
   The if statement is used to execute a block of code only if a specified condition is true. It enables decision-making in a program, allowing the program to respond differently based on varying conditions.
   For example: -
   It can be used to check user input, validate data, or control the flow of execution based on dynamic conditions.
   Syntax: -
   ```
   if (condition) {
      // code to be executed if condition is true
   }
   ```
5. ***Define selection control statement?***
   A selection control statement allows a program to choose different paths of execution based on the evaluation of one or more conditions.
   These statements enable decision-making in the code, directing the program to execute specific blocks of code while skipping others based on the conditions.
   For examples: -
   Selection control statements include if, if-else, and switch statements.
6. ***Define repetition control statement?***
   A repetition control statement, also known as a loop, enables a block of code to be executed repeatedly based on a condition. This allows for tasks to be repeated multiple times without the need to write the same code over and over.
   For example: -
   Common repetition control statements include for, while, and do-while loops

7. How many types of selection control statement?

There are mainly types of selection control statement are as follow: -

1. *If*
2. *If-else*
3. *If-else-if*
4. *Nested-if*
5. *Switch*

8. Differentiate between if and if-else statement?

if statement

Executes a block of code if a specified condition is true. If the condition is false, the block of code is skipped entirely.

Syntax

```
if (condition) {
  // code to be executed if condition is true
}
```

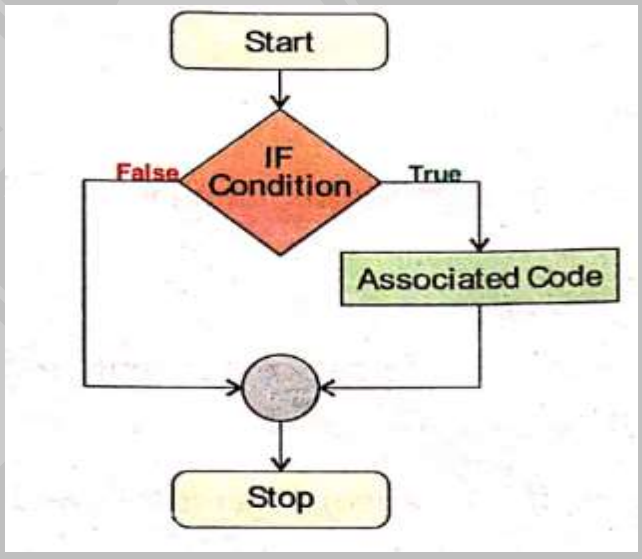
if-else statement

Executes one block of code if a specified condition is true and another block of code if the condition is false.

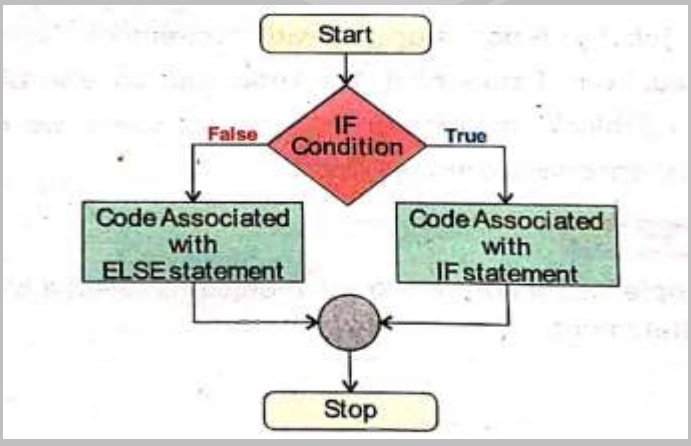
Syntax

```
if (condition) {
  // code to be executed if condition is true
} else {
  // code to be executed if condition is false
}
```

9. Draw the flowchart of if statement?



10. Draw the flowchart of if-else statement?



11. Write down the syntax/structure of if-else statement?

Syntax

```
if (condition) {
   // code to be executed if condition is
true
} else {
   // code to be executed if condition is
false
}
```

12. Define compound statement?

A compound statement in C (and many other programming languages) is a group of statements enclosed in curly braces {}. It allows multiple statements to be treated as a single statement in the syntax of the language.

For example:

```
{
   int x = 5;
   x++;
   printf("%d\n", x);
}
```

13. Define Nested Coding condition?

A condition inside and another condition is called Nested Selection Coding. If the outer condition is true then the inner condition is ***on*** *and the body will be executed.*

14. Why if-else statement used in C language?

The if-else statement in C is used for conditional execution of code. It allows the program to execute a particular block of code if a specified
condition is true and another block of code
if the condition is false. This is essential for making decisions within a program and controlling the flow of execution based on different conditions.

For example: -

```
int a = 10, b = 20;
if (a > b) {
   printf("a is greater than b");
} else {
   printf("a is not greater than b");
}
```

15. How can u close if and if-else statement?

In C, if and if-else statements are closed by ending their respective code blocks, typically using curly braces {}.

For a simple if statement:

```
if (condition) {
   // code to execute if condition is true
}
```

For an if-else statement:

```
if (condition) {
   // code to execute if condition is true
} else {
   // code to execute if condition is false
}
```

16. What is the common mistake in the compound statement?

A common mistake in using compound statements is omitting the curly braces {} when they are actually needed. This can lead to unexpected behavior, especially in control flow statements like if-else and loops.

For example:

```
if (condition)
   statement1;
   statement2;  // This statement is always executed regardless of the condition
```

17. Define condition?

In programming, a condition is an expression that evaluates to either true or false. Conditions are used in control flow statements to determine which code block should be executed. Conditions often involve comparison operators (e.g., ==, !=, <, >, <=, >=) and logical operators (e.g., &&, ||, !).

For example: -

```
int a = 10;
int b = 20;
if (a < b) {
   printf("a is less than b");
}
```

18. Write down the syntax of if-else-if statement?

The syntax of an if-else-if statement in C is:

Syntax: -

```
if (condition1) {
   // code to execute if condition1 is true
} else if (condition2) {
   // code to execute if condition2 is true
} else if (condition3) {
   // code to execute if condition3 is true
} else {
   // code to execute if none of the conditions above are true
}
```

19. What do we need selection statement?

Selection statements are needed in programming to control the flow of execution based on conditions. They allow the program to make decisions and execute different blocks of code depending on whether a condition is true or false. This is essential for creating dynamic and responsive programs that can handle various scenarios and inputs.

20. What is the use of nested selection structure?

A nested selection structure is used when you need to make multiple decisions within another decision. It allows more complex decision-making processes where each branch of a selection statement can itself be another selection statement. This enables the creation of more intricate and nuanced control flows within a program

21. Differentiate between nested if & if-else-if statement?

Nested if:

- *A nested if statement is an if statement within another if or else statement.*
- *Each if statement can have its own separate conditions and blocks of code.*

Example: -

```
if (condition1) {
    if (condition2) {
        // code to execute if both condition1 and condition2 are true
    }
}
```

If-else-if:

- *An if-else-if statement is a chain of conditions where each condition is checked only if the previous conditions were false.*
- *It is used to handle multiple conditions that are mutually exclusive.*

Example: -

```
if (condition1) {
    // code to execute if condition1 is true
} else if (condition2) {
    // code to execute if condition1 is false and condition2 is true
} else {
    // code to execute if none of the conditions are true
}
```

22. Write down the syntax of if-else-if statement?

The syntax for an if-else-if statement is:

```
if (condition1) {
    // code to execute if condition1 is true
} else if (condition2) {
    // code to execute if condition1 is false and condition2 is true
} else if (condition3) {
    // code to execute if condition1 and condition2 are false and condition3 is true
} else {
    // code to execute if none of the conditions are true
}
```

23. Write down the syntax of nested if statement?

The syntax for an if-else-if statement is:

```
if (condition1) {
    // code to execute if condition1 is true
    if (condition2) {
        // code to execute if both condition1 and condition2 are true
    } else {
        // code to execute if condition1 is true and condition2 is false
    }
} else {
    // code to execute if condition1 is false
}
```

LONG QUESTIONS

1. Write a note on If-else-If Selection Structure?

If-else-if: -

- *An if-else-if statement is a chain of conditions where each condition is checked only if the previous conditions were false.*
- *It is used to handle multiple conditions that are mutually exclusive.*

Syntax: -

The syntax for an if-else-if statement is:

```
if (condition1) {
    // code to execute if condition1 is true
} else if (condition2) {
    // code to execute if condition1 is false and condition2 is true
} else if (condition3) {
    // code to execute if condition1 and condition2 are false and condition3 is true
} else {
    // code to execute if none of the conditions are true
}
```

2. *Write down the structure of if-else statement with brief description?*

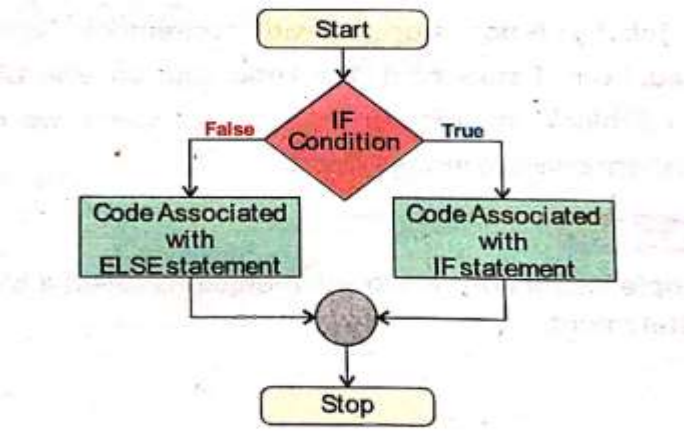
If-else:

Executes one block of code if a specified condition is true and another block of code if the condition is false.

Syntax

```
if (condition) {
   // code to be executed if condition is true
} else {
   // code to be executed if condition is false
}
```

Flowchart: -



How can u close if and if-else statement?

In C, if and if-else statements are closed by ending their respective code blocks, typically using curly braces {}.

For a simple if statement:

```
if (condition) {
   // code to execute if condition is true
}
```

For an if-else statement:

```
if (condition) {
   // code to execute if condition is true
} else {
   // code to execute if condition is false
}
```

Why if-else statement used in C language?

The if-else statement in C is used for conditional execution of code. It allows the program to execute a particular block of code if a specified
condition is true and another block of code
if the condition is false. This is essential for making decisions within a program and controlling the flow of execution based on different conditions.

For example: -

```
int a = 10, b = 20;
if (a > b) {
   printf("a is greater than b");
} else {
   printf("a is not greater than b");
}
```

3. *Write a note on Selection Structure?*

Selection control statement: -

A selection control statement allows a program to choose different paths of execution based on the evaluation of one or more conditions. These statements enable decision-making in the code, directing the program to execute specific blocks of code while skipping others based on the conditions.

Types: -

Selection control statements include if, if-else, if-else-if, nested if and switch statements

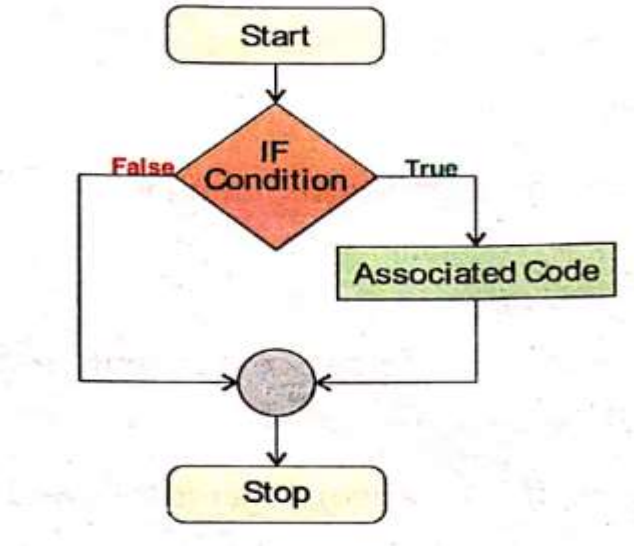
1. *If condition: -*

Executes a block of code if a specified condition is true. If the condition is false, the block of code is skipped entirely.

Syntax

```
if (condition) {
   // code to be executed if condition is true
}
```

Flowchart



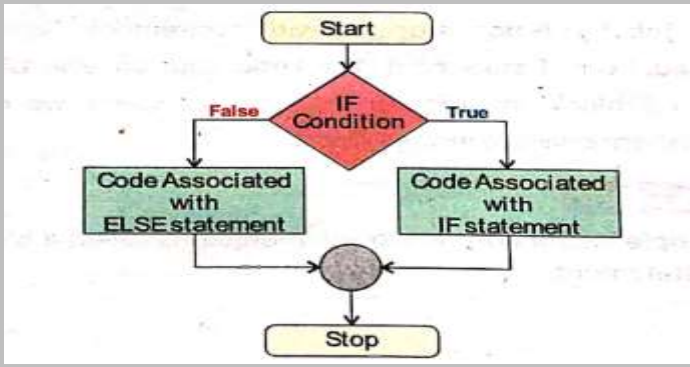
2. *If-else:*

Executes one block of code if a specified condition is true and another block of code if the condition is false.

Syntax

```
if (condition) {
   // code to be executed if condition is true
} else {
   // code to be executed if condition is false
}
```

Flowchart: -



3. *If-else-if: -*

- *An if-else-if statement is a chain of conditions where each condition is checked only if the previous conditions were false.*
- *It is used to handle multiple conditions that are mutually exclusive.*

Syntax: -

The syntax for an if-else-if statement is:

```
if (condition1) {
   // code to execute if condition1 is true
} else if (condition2) {
   // code to execute if condition1 is false and condition2 is true
} else if (condition3) {
   // code to execute if condition1 and condition2 are false and condition3 is true
} else {
   // code to execute if none of the conditions are true
}
```

4. *Nested if: -*

*A condition inside and another condition is called Nested Selection Coding. If the outer condition is true then the inner condition is **on** and the body will be executed. Each if statement can have its own separate conditions and blocks of code.*

Syntax: -

```
    if (condition1) {
       if (condition2) {
          // code to execute if both condition1 and condition2 are true
       }
    }
```

Needs: -

Selection statements are needed in programming to control the flow of execution based on conditions. They allow the program to make decisions and execute different blocks of code depending on whether a condition is true or false. This is essential for creating dynamic and responsive programs that can handle various scenarios and inputs

Purpose: -

A nested selection structure is used when you need to make multiple decisions within another decision. It allows more complex decision-making processes where each branch of a selection statement can itself be another selection statement. This enables the creation of more intricate and nuanced control flows within a program

4. *Write a note on Control Statement?*

Controlling statements are programming constructs that control the flow of execution of statements in a program based on certain conditions or loops. They determine the order in which instructions are executed. There are three types of controlling statement are as follow: -

1. *Sequential control statement*
   A sequential control statement refers to the default mode of execution in a program where statements are executed one after the other in the exact order they appear. Each statement runs exactly once, ensuring a linear and predictable flow of execution
2. *Selection control statement*
   A selection control statement allows a program to choose different paths of execution based on the evaluation of one or more conditions. These statements enable decision-making in the code, directing the program to execute specific blocks of code while skipping others based on the conditions.
   Types: -
   Selection control statements include if, if-else, if-else-if, nested if and switch statements
3. *Repetition control statement*
   A repetition control statement, also known as a loop, enables a block of code to be executed repeatedly based on a condition. This allows for tasks to be repeated multiple times without the need to write the same code over and over.
   Types: -
   Repetition control statements include for, while, and do-while, for loops and for each loop

5. *Write a note on Nested if?*

*A condition inside and another condition is called Nested Selection Coding. If the outer condition is true then the inner condition is **on** and the body will be executed. Each if statement can have its own separate conditions and blocks of code.*

Syntax: -

```
if (condition1) {
   if (condition2) {
      // code to execute if both condition1 and condition2 are true
   }
}
```

Needs: -

Selection statements are needed in programming to control the flow of execution based on conditions. They allow the program to make decisions and execute different blocks of code depending on whether a condition is true or false. This is essential for creating dynamic and responsive programs that can handle various scenarios and inputs

Purpose: -

A nested selection structure is used when you need to make multiple decisions within another decision. It allows more complex decision-making processes where each branch of a selection statement can itself be another selection statement. This enables the creation of more intricate and nuanced control flows within a program

LONG CODING QUESTIONS

1. ***Write a program that takes the total bill as input and tell how much discount the user has got and what is the discounted price?***

Total bills	*Discount*
1000	*10%*
2500	*20%*
5000	*35%*
10000	*50%*

Code: -

```
#include <stdio.h>
using namespace std;
int main() {
   float totalBill, discount, discountedPrice;
   printf("Enter the total bill: ");
   scanf("%f", &totalBill);
   if (totalBill >= 10000) {
      discount = totalBill * 0.50;
   } else if (totalBill >= 5000) {
      discount = totalBill * 0.35;
   } else if (totalBill >= 2500) {
      discount = totalBill * 0.20;
   } else if (totalBill >= 1000) {
      discount = totalBill * 0.10;
   } else {
      discount = 0;
   }
   discountedPrice = totalBill - discount;
   printf("Discount: %.2f\n", discount);
   printf("Discounted Price: %.2f\n", discountedPrice);
   return 0;
}
```

2. ***Write a program that takes as input the original price and sale price of a product and tells whether the product is sold on profit or loss. The program should also tell the profit/loss percentage.***

Code: -

```
#include <stdio.h>
using namespace std;
int main() {
   float originalPrice, salePrice, profitLoss, profitLossPercentage;
   printf("Enter the original price: ");
   scanf("%f", &originalPrice);
   printf("Enter the sale price: ");
   scanf("%f", &salePrice);
   if (salePrice > originalPrice) {
      profitLoss = salePrice - originalPrice;
      profitLossPercentage = (profitLoss / originalPrice) * 100;
      printf("Profit: %.2f\n", profitLoss);
      printf("Profit Percentage: %.2f%%\n", profitLossPercentage);
   } else if (salePrice < originalPrice) {
      profitLoss = originalPrice - salePrice;
      profitLossPercentage = (profitLoss / originalPrice) * 100;
      printf("Loss: %.2f\n", profitLoss);
      printf("Loss Percentage: %.2f%%\n", profitLossPercentage);
   }else {
      printf("No Profit, No Loss\n");
   }
   return 0;
}
```

3. ***Write a program that takes as input, the obtained mark and total marks of matric, intermediate and entrance test. The program should tell whether the student is eligible or not: -***
   - ***At least 60% in matric***
   - ***At least 65% in intermediate in (FSC pre-engineering & ICS)***
   - ***At least 65% in entrance test***

Code: -

```
#include <stdio.h>
using namespace std;
int main() {
   float obtainedMatric, totalMatric, obtainedInter, totalInter, obtainedEntrance, totalEntrance;
   float percentageMatric, percentageInter, percentageEntrance;
   printf("Enter obtained marks in Matric: ");
   scanf("%f", &obtainedMatric);
   printf("Enter total marks in Matric: ");
   scanf("%f", &totalMatric);
   printf("Enter obtained marks in Intermediate: ");
   scanf("%f", &obtainedInter);
   printf("Enter total marks in Intermediate: ");
   scanf("%f", &totalInter);
   printf("Enter obtained marks in Entrance test: ");
   scanf("%f", &obtainedEntrance);
   printf("Enter total marks in Entrance test: ");
   scanf("%f", &totalEntrance);
   percentageMatric = (obtainedMatric / totalMatric) * 100;
   percentageInter = (obtainedInter / totalInter) * 100;
   percentageEntrance = (obtainedEntrance / totalEntrance) * 100;
   if (percentageMatric >= 60 && percentageInter >= 65 && percentageEntrance >= 65) {
      printf("The student is eligible.\n");
   } else {
      printf("The student is not eligible.\n");
   }
   return 0;
}
```

4. ***Write a program that calculate the bonus an employee and get on the following basis:***

Salary	***Experience With Compa***	***Bonus take***	***Bonus***
10000	***2 year***	***5***	***1500***
10000	***3 year***	***10***	***2500***
25000	***3 year***	***4***	***2000***
75000	***4 year***	***7***	***3500***
100000	***5 year***	***10***	***5000***

The program should take as input, the salary, experience and number of bonus takes of the employees. The program should display the bonus on the screen

Code: -

```
#include <stdio.h>
using namespace std;
int main() {
   float salary, bonus = 0;
   int experience, bonusTakes;
   printf("Enter salary: ");
   scanf("%f", &salary);
   printf("Enter experience with company (in years): ");
   scanf("%d", &experience);
   printf("Enter number of bonus takes: ");
   scanf("%d", &bonusTakes);
   if (salary == 10000 && experience == 2 && bonusTakes == 5) {
      bonus = 1500;
```

```
    } else if (salary == 10000 && experience == 3 && bonusTakes == 10) {
       bonus = 2500;
    } else if (salary == 25000 && experience == 3 && bonusTakes == 4) {
       bonus = 2000;
    } else if (salary == 75000 && experience == 4 && bonusTakes == 7) {
       bonus = 3500;
    } else if (salary == 100000 && experience == 5 && bonusTakes == 10) {
       bonus = 5000;
    } else {
       printf("No bonus applicable for the given inputs.\n");
       return 0;
    }
    printf("The bonus is: %.2f\n", bonus);
    return 0;
}
```

5. ***Write a program that find and display the area of a triangle, parallelogram, rhombus of trapezium according to the choice of user.***

Code: -

Switch Statement

```
#include<stdio.h>
using namespace std;
int main(){
    int choice;
    float base, height, base1, base2, diagonal1, diagonal2, area;
    float areaoftriangle, areaofparallelogram, areaofrhoumbous, areaoftraphezium;
//  Display the menu
    printf("Choose any shape on your mind(1-4):\n");
    printf("1. Triangle:\n");
    printf("2. Parallelogram:\n");
    printf("3. Rhoumbous:\n");
    printf("4. Traphezium:\n");
    scanf("%d",&choice);
    switch(choice){
        case 1:
            printf("Enter the base of a Traingle:\t");
            scanf("%f",&base);
            printf("Enter the height of a Triangle:\t");
            scanf("%f",&height);
            areaoftriangle=0.5*base*height;
            printf("The area of a Traingle is:%.2f\t",areaoftriangle);
        break;
        case 2:
            printf("Enter the base of a Parallelogram:\t");
            scanf("%f",&base);
            printf("Enter the height of a Parallelogram:\t");
            scanf("%f",height);
            areaofparallelogram=base*height;
            printf("The area of a Parallelogram is:%2fn\t",areaofparallelogram);
        break;
        case 3:
            printf("Enter the 1st diagonal of Rhoumbous:\t");
            scanf("%f",&diagonal1);
            printf("Enter the 2nd diagonal of Rhoumbous:\t");
            scanf("%f",&diagonal2);
            areaofrhoumbous=0.5*diagonal1*diagonal2;
            printf("The area of a Rhoumbous is:%2fn\t",areaofrhoumbous);
        break;
```

```
            case 4:
                printf("Enter the 1st base of Traphezium:\t");
                scanf("%f",&base1);
                printf("Enter the 2nd base of Traphezium:\t");
                scanf("%f",&base2);
                printf("Enter teh height od Traphezium:\t");
                scanf("%f",&height);
                areaoftraphezium=0.5*base1*base2*height;
                printf("The area of a Traphezium is:%2f\t",areaoftraphezium);
            break;
            default:
                printf("@@@@@Invalid choice@@@@@");
            break;
    }
    return 0;
}
```

If else if statement

```
#include<stdio.h>
using namespace std;
int main(){
    int choice;
    float base, height,base1,base2,diagonal1,diagonal2,area;
    float areaoftriangle,areaofparallelogram,areaofrhoumbous,areaoftraphezium;
//  Display the menu
    printf("Choose any shape on your mind(1-4):\n 1.Triangle\n 2.Parralellogram\n 3.Rhoummbous\n 4.Traphezium
\n");
    scanf("%d",&choice);
    if(choice==1){
        printf("Enter the base of a Traingle:\t");
        scanf("%f",&base);
        printf("Enter the height of a Triangle:\t");
        scanf("%f",&height);
        areaoftriangle=0.5*base*height;
        printf("The area of a Traingle is:%.2f\t",areaoftriangle);
    }else if(choice==2){
        printf("Enter the base of a Parallelogram:\t");
        scanf("%f",&base);
        printf("Enter the height of a Parallelogram:\t");
        scanf("%f",height);
        areaofparallelogram=base*height;
        printf("The area of a Parallelogram is:%2fn\t",areaofparallelogram);
    }else if(choice==3){
        printf("Enter the 1st diagonal of Rhoumbous:\t");
        scanf("%f",&diagonal1);
        printf("Enter the 2nd diagonal of Rhoumbous:\t");
        scanf("%f",&diagonal2);
        areaofrhoumbous=0.5*diagonal1*diagonal2;
        printf("The area of a Rhoumbous is:%2fn\t",areaofrhoumbous);
    }else if(choice==4){
        printf("Enter the 1st base of Traphezium:\t");
        scanf("%f",&base1);
        printf("Enter the 2nd base of Traphezium:\t");
        scanf("%f",&base2);
        printf("Enter teh height od Traphezium:\t");
        scanf("%f",&height);
        areaoftraphezium=0.5*base1*base2*height;
        printf("The area of a Traphezium is:%2f\t",areaoftraphezium);
    }else{
```

```
            printf("@@@@@ Invalid Choice @@@@@");
        }
        return 0;
    }
```

6. ***The eligibility criteria of a university for its different undergraduate program is as follow:***
   BSSE: 80% more than in intermediate
   BSCS: 75% more than in intermediate
   BSIT: 70% more than in intermediate

Otherwise the university do not enroll the student in any programs, write a program takes the percentage of intermediate and tell for which the program the student is eligible to apply.

Code: -

```
#include <stdio.h>
Using namespace std;
int main() {
   float percentage;
   printf("Enter the intermediate percentage: ");
   scanf("%f", &percentage);
   if (percentage >= 80) {
      printf("The student is eligible for BSSE.\n");
   }
        else if (percentage >= 75) {
      printf("The student is eligible for BSCS.\n");
   }
        else if (percentage >= 70) {
      printf("The student is eligible for BSIT.\n");
   }
        else {
      printf("The student is not eligible for any program.\n");
   }
   return 0;
}
```

7. ***Write a program that takes two input as integers and ask the user to enter the choice 1 to 4. The program should perform the operation according to the given table and display the result:***

Choice	Operation
1	Addition
2	Subtraction
3	Multiplication
4	Division

Code: -

```
#include<stdio.h>
using namespace std;
int main(){
        int choice,num1,num2,result;
        float resultdiv;
        printf("Enter your operation number(1-4)\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n");
        scanf("%d",&choice);
        printf("Enter the first & second numbers=\n");
        scanf("%d%d",&num1,&num2);
        if(choice==1){
                result=num1+num2;
                printf("The result of Addition is %d",result);
        }else if(choice==2){
                result=num1-num2;
                printf("The result of Subtraction is %d",result);
        }else if(choice==3){
```

```
            result=num1*num2;
            printf("The result of Multiplication is %d",result);
        }else if(choice==4){
            if(num2!=0){
                resultdiv = (float)num1 / num2;
                printf("It is not a zero value of num2",resultdiv);
            }else{
                printf("The value of num2 is zero",resultdiv);
            }
        }
        return 0;
}
```

1. Write a function that takes two input as integers and ask the user to enter the choice 1 to 4. The program should perform the operation according to the given table and display the result:

Choice	*Operation*
1	*Addition*
2	*Subtraction*
3	*Multiplication*
4	*Division*

Code: -

```
#include <stdio.h>
using namespace std;
// Function prototype
void performOperation(int choice, int num1, int num2);
int main() {
  int choice, num1, num2;
  // Input two integers
  printf("Enter the first integer: ");
  scanf("%d", &num1);
  printf("Enter the second integer: ");
  scanf("%d", &num2);
  // Display the menu
  printf("1. Addition\n");
  printf("2. Subtraction\n");
  printf("3. Multiplication\n");
  printf("4. Division\n");
  printf("Choose an operation (1-4): ");
  scanf("%d", &choice);
  // Perform the operation based on user choice
  performOperation(choice, num1, num2);
  return 0;
}
// Function definition
void performOperation(int choice, int num1, int num2) {
  int result;
  float resultDiv;
  switch(choice){
        case 1:
     result = num1 + num2;
     printf("The result of addition is: %d\n", result);
```

```
    break;
    case 2:
      result = num1 - num2;
      printf("The result of subtraction is: %d\n", result);
    break;
            case 3:
      result = num1 * num2;
      printf("The result of multiplication is: %d\n", result);
    break;
    case 4:
      if(num2 != 0){
        resultDiv = (float)num1 / num2;
        printf("The result of division is: %.2f\n", resultDiv);
        }else{
          printf("Division by zero is not allowed.\n");
        }
    break;
    default:
            printf("Invalid choice.\n");
    break;
  }
}
```

ERRORS & OUTPUT

Q4. Identify errors in the following code segments. Assume that variables have already been declared.

a)
```
if (x = 10)
        printf ("Good");
```
Error: Invalid sign = in C language.

b)
```
if (a < b && b < c);
        sum = a + b + c;
else
        multiply = a * b * c;
```
Error: ; is not used in if statement (Syntax Error).

c)
```
if (a < 7 < b)
        printf ("7");
```
Error: 7 is printing as string.

d)
```
if (a == b &| x == y)
        flag = true;
else
        flag = false;
```
Error: Invalid syntax of AND and OR operator (&|)

e)
```
if (sum == 60 | | product == 175)
        printf ("Accepted %c), sum);
else
        if (sum >=45 || product > 100)
        printf ("Considered %d" , sum);
else
        printf ("Rejected")
```
Error: Double quotes are missing at printf ("Accepted %c), sum); and %d should be used instead of %c.

Q5 Write down output of the following code segments.

a)
```
int a = 7, b = 10;
a = a + b;
if ( a> 20 && b < 20 )
        b = a + b;
printf ("a = %d, b = %d", a, b);
```

Output: a=17, b=10

b)
```
int x = 45;
if (x + 20 * 7 == 455)
        printf ("Look's Good");
else
        printf ("Hope for the Best");
```

Output: Hope for the Best

c)
```
char c1 = 'Y', c2 = 'N';
int n1 = 5, n2 = 9;
n1 = n1 + 1;
c1 = c2;
if (n1 == n2 && c1 == c2)
        printf ("%d = %d and %c = %c" , n1, n2, c1, c1);
else
        if (n1 < n2 && c1 == c2)
        printf ("%d < %d and %c = %c", n1, n2, c1, c2);
else
        printf ("Better Luck Next Time!");
```

Output: 6<9 and N=N

d)
```
int a = 34, b = 32, c = 7, d = 15;
a = b + c + d;
if (a < 100)
a = a * 2;
b = b * c;
c = c + d;
if ( a > b && c == d)
        {
                c = d;
                b = c;
                a = b;
        }
else if (a > b &&  c > d || b >= d +c)
        {
                d = c * c;
                a = b * b;
        }
printf ("a=%d, b=%d, c=%d, d=%d", a, b, c, d);
```

Output: a=15360, b=224, c=22, d=484

e)
```
int x = 5, y = 7, z = 9;
if (x % 2 == 0)
  x++;
else
    x = y+ z;
    printf (" x = %d\n", x);
    if ( x % 2 == 1 && y % 2 == 1 && z % 2 == 1)
            printf ("All are Odd");
    if ( x> y || x < z)
    {
            if ( x > y )
                    y++;
            else
                    if (x < z )
                    x++;
    }
    printf ("x = %d, y = %d, z = %d", x, y, z);
```

Output: x=6, y=7, z=9

MCQ'S (MULTIPLE CHOICE QUESTION'S)

Q1 Multiple Choice Questions

Sr. No	Question	A	B	C	D
1	Conditional logic helps in_____	Decisions	iterations	traversing	all
2	____statements describe the sequence in which statements of the program should be executed.	Loop	Conditional	Control	All
3	In if statement, what happens if condition is false?	Program crashes	Index out of bound error	Further code executes	Compiler asks to change condition

4	int a =5;
	if (a < 10)
	a++;
	else
	if (a > 4)
	a - -;
	Which of the following statements will execute?
		a + +;	a--;	both (a) and (b)	None

5	Which of the following is the condition to check a is a factor of c?	a % c == 0	c % a ==0	a* c== 0;	a+c==0
6	A condition can be any___ expression.	arithmetic	relational	logical	arithmetic, relational or logical
7	An if statement inside another if statement is called_____ structure.	Nested	boxed	repeated	decomposed
8	A set of multiple instructions enclosed in braces is called a _____	Box	List	block	job

Key

1	2	3	4	5	6	7	8
A	C	C	A	A	D	A	C

MCQS

Sr. No	Question	A	B	C	D
1	We can control the flow of program execution through_____.	control statements	sequential control statements	selection control statements	repetition control statements
2	There are _____ types of control statements in C language.	2	3	5	7
3	Which one from the following is not a type of control statement?	sequential control statements	selection control statements	repetition control statements	check control statement
4	The statements which help us to decide which statements should be executed next, on the basis of conditions, are called_____.	repetition control statements	sequential control statements	Selection Statements	check control statement
5	Types of selection statements are:	if statement	if-else statement	both A and B	nested statement
6	C language provides _____ in which we specify a condition, and associate a code to it.	repetition control statements	if statement	if-else statement	selection control statements
7	The code gets executed if the specified condition turns out to be_____.	False	True	both A and B	none
8	The structure of if statement is:	if (condition) Associated Code	(condition) Associated Code	else (condition) Associated Code	if else (condition)
9	In the structure of if statement, if is a keyword that is followed by a condition inside_____.	( )	< >	{ }	[ ]
10	According to the_____, all the statements are executed in the given sequence.	selection control	repetition control	sequential control	relational control
11	_____is the default control structure in C language.	Check control	Sequential control	Relational control	All
12	__executes the set of statements under if statement if the condition is true.	Control statement	Condition statement	if statement	if-else statement
13	A set of multiple instructions enclosed in braces is called a ____or a_____.	block, compound statement	block, relational statement	code; block	compound statement, code
14	If there are more than one instructions under *if statement* or *else statement*, enclose them in the form of a_____.	Box	Braces	block	code
15	Conditional statements within conditional statements are called____	Conditional structure	nested selection structures	if else structure	if nested selection

16	In_____, it is a common mistake to omit one or two braces while typing	compound statements	control statement	selection statement	relational statement

KEY

1	2	3	4	5	6	7	8	9	10
A	B	D	C	C	B	B	A	A	C
11	12	13	14	15	16				
B	D	A	C	B	A				