Day 2: Conditional Statements (4-8-2025)

1. **Write a program to check if a number is positive, negative, or zero.**

IPO:

INPUT: Take an integer value say num
PROCESS:

If num > 0 => Positive
If num < 0 => Negative
If num == 0 => Zero

OUTPUT: Display whether the number is positive, negative, or zero.

CODE:

```
#include<stdio.h>
void main()
{
	int num;
	scanf("%d", &num);
	if (num > 0)
	printf("The number is positive.");
	else if (num < 0)
	printf("The number is negative.");
	else
	printf("The number is zero.");
}
```

OUTPUT;

```
Output
-15
The number is negative.
```

2.Write a program to find the largest among three numbers

IPO

INPUT: Three numbers (`a`, `b`, `c`)
PROCESS: Compare using if-else to find the largest
OUTPUT: Display the largest number

CODE;

```
#include<stdio.h>
void main()
{
	int a, b, c;
	scanf("%d %d %d", &a, &b, &c);
	 if (a>= b && a>= c)
		printf("%d is the largest.", a);
	else if (b>= c)
		printf("%d is the largest.", b);
	else
		printf("%d is the largest.", c);
}
```

OUTPUT;

```
Output
12 13 11
13 is the largest.
```

3.Write a program to check if a year is a leap year.

IPO

INPUT: A year say ‘year’

PROCESS:

If divisible by 400 → Leap

Else if divisible by 4 → Leap

OUTPUT: Whether it's a leap year or not

CODE:

```
#include<stdio.h>
void main()
{
   int year;
   scanf("%d", &year);
   if ((year % 400 == 0) || (year % 4 == 0))
      printf("%d is a leap year.", year);
   else
      printf("%d is not a leap year.", year);
}
```

OUTPUT;

```
Output
2024
2024 is a leap year.
```

4.Write a program to check whether a character is a vowel or consonant

IPO

INPUT: A single character say ‘ch’

PROCESS:

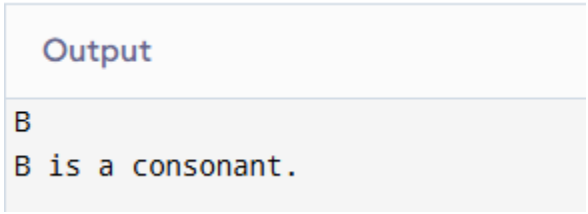
Check if ch is a, e, i, o, u

OUTPUT: Display vowel or consonant

CODE:

```
#include<stdio.h>
void main()
{
   char ch;
   scanf(" %c", &ch);
   if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == 'A' ||
      ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
      printf("%c is a vowel.", ch);
   else
      printf("%c is a consonant.", ch);
}
```

OUTPUT;



5.Write a program to assign grades based on marks.

IPO

INPUT: Marks out of 100 (`marks`)
PROCESS:
Use `if-else` ladder to assign grade
OUTPUT: Grade A, B, C, D, or Fail

CODE;

```
#include<stdio.h>
void main()
{
   int marks;
   scanf("%d", &marks);
   if (marks >= 90)
      printf("Grade A");
   else if (marks >= 75)
      printf("Grade B");
   else if (marks >= 60)
      printf("Grade C");
   else if (marks >= 40)
      printf("Grade D");
```

```
    else
        printf("Fail");
}
```

OUTPUT;

```
Output

94
Grade A
```

6.Write a program to check whether a number is divisible by 5 and 11.

IPO

INPUT: An integer (`num`)

PROCESS: Check `num % 5 == 0 && num % 11 == 0`

OUTPUT: Whether divisible or not

CODE;

```
#include<stdio.h>
void main()
{
   int num;
   scanf("%d", &num);
   if (num % 5 == 0 && num % 11 == 0)
      printf("%d is divisible by 5 and 11.", num);
   else
      printf("%d is not divisible by 5 and 11.", num);
}
```

OUTPUT;

```
Output

55
55 is divisible by  5 and 11.
```

7.Write a program to find the absolute value of a number

IPO

INPUT: A number (`val`)
PROCESS: If `val < 0`, change sign
OUTPUT: Absolute value

CODE;

```
#include<stdio.h>
void main()
{
   int val;
   scanf("%d", &val);
   if (val < 0 || val>0)
       printf("Absolute value = %d", val);
}
```

OUTPUT;

```
Output

5
Absolute value = 5
```

8.Write a menu-driven program to perform +, -, *, / operations.

IPO

INPUT: Two numbers and operation choice
PROCESS: Perform operation based on choice
OUTPUT: Result of the selected operation

CODE;

```
#include<stdio.h>
void main()
{
   int a, b, choice;
   scanf("%d %d", &a, &b);
   scanf("%d", &choice);
   switch(choice)
   {
      case 1: printf("Sum = %d", a + b); break;
      case 2: printf("Difference = %d", a - b); break;
      case 3: printf("Product = %d", a * b); break;
      case 4:printf("Quotient = %d", a / b);
   }
}
```

OUTPUT;

```
Output
10 5 2
Difference = 5
```

```
Output
10 5 3
Product = 50
```

```
Output
10 5 4
Quotient = 2
```

9.Write a program to find roots of a quadratic equation.

IPO

INPUT: Coefficients a, b, c

PROCESS: Use formula:

Discriminant $d = b^2 - 4ac$

Roots based on d : real & distinct / equal / imaginary

OUTPUT: Display roots

CODE;

```
#include<stdio.h>
#include<math.h>
void main()
{
   float a, b, c, d, root1, root2;
```

```
    scanf("%f %f %f", &a, &b, &c);
    d = b * b - 4 * a * c;
    root1 = (-b + sqrt(d)) / (2 * a);
    root2 = (-b - sqrt(d)) / (2 * a);
    printf("Root 1 = %f\n", root1);
    printf("Root 2 = %f", root2);
}
```

OUTPUT;

```
Output

1 5 6
Root 1 = -2.000000
Root 2 = -3.000000
```

10.Write a program to find the number of digits in a number

IPO

INPUT: A positive integer (`num`)

PROCESS: Use loop: `num /= 10` until 0

OUTPUT: Number of digits

CODE;

```
#include<stdio.h>
void main()
{
    int num, count = 0;
    scanf("%d", &num);
    if(num == 0)
```

```
        count = 1;
    else
    {
        while(num != 0)
        {
            num = num / 10;
            count++;
        }
    }
    printf("Number of digits = %d", count);
}
```

OUTPUT;

```
Output

123
Number of digits = 3
```