ASSIGNMENT-1

FOR CODING IN C

NAME-RAMAN KUMAR

COURSE-B.TECH(HONS.)CSE

SEMESTER -1

BATCH NO-22

YEAR 2023-2027

SAP ID-500120223

ROLL NO-R2142230699

TEACHER NAME-DR. TARANDEEP KAUR BHATIA

_____

DATE- 23 AUGUST 2023

TEACHER'S SIGNATURE

Q1.) WHAT IS C PROGRAMMING LANGUAGE?

WRITE THE FEATURES OF C.

→ **C is considered a structured programming language that supports both low-level and high-level programming, making it adaptable for both system and application-level programming. It is also recognized as a middle-level language that combines the attributes of both assembly-level and high-level languages.**

→**The C programming language possesses numerous features that make it a prime option for creating system-level software applications.**

1. **Modularity → Modularity is a key feature of C that enables developers to write code in small, reusable modules. This makes it simpler to maintain and debug the code.**
2. **Efficiency → C is a programming language that is highly esteemed for its remarkable efficiency. It is capable of running at a low level and granting developers an unparalleled level of control over system resources.**
3. **Portability → The programming language C provides developers with the ability to write code that operates at a low level, which grants them greater influence over the hardware that the code is running on.**
4. **Rich Library Support → C, the programming language, has a large library of pre-existing functions that can be used to help create robust software.**
5. **Memory Management → C provides direct access to memory through pointers, enabling efficient manipulation and allocation of memory.**

Q2.) WAP TO SWAP TWO NO BY USING THIRD VARIABLE.

→**PROGRAM:**

```
  #include <stdio.h>

int main() {

   int num1, num2, temp;

    // Input the two numbers

  printf("Enter the first number , second number\n : ");

 scanf("%d,%d",&num1,&num2);

    // Swap the numbers using a temporary variable

    temp = num1;

    num1 = num2;

    num2 = temp;

    printf("After swapping:\n");

    printf("First number: %d\n", num1);

    printf("Second number: %d\n", num2);

    return 0;

}
```

OUTPUT:

c programming > C assignmentq1.c > main()

```c
#include <stdio.h>
int main() {
   int num1, num2, temp;
    // Input the two numbers
  printf("Enter the first number , second number  : ");
 scanf("%d,%d",&num1,&num2);
    // Swap the numbers using a temporary variable
    temp = num1;
    num1 = num2;
    num2 = temp;
    printf("After swapping:\n");
    printf("First number: %d\n", num1);
    printf("Second number: %d\n", num2);
    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ssignmentq1 } ; if ($?) { .\assignmentq1 }
Enter the first number , second number  : 6,5
After swapping:
First number: 5
Second number: 6
```

Q3.) WAP TO GET ALL THE ROOTS OF A QUADRATIC EQUATION.

→PROGRAM:

```
#include <stdio.h>
#include <math.h>
int main() {
double a, b, c, discriminant, root1, root2;
 // Input coefficients from the user
printf("Enter coefficients of the quadratic equation (a, b, c): ");
scanf("%lf %lf %lf", &a, &b, &c);
// Calculate the discriminant
discriminant = b * b - 4 * a * c;
// Check the discriminant to determine the nature of the roots
if (discriminant > 0) {
// Two real and distinct roots
root1 = (-b + sqrt(discriminant)) / (2 * a);
root2 = (-b - sqrt(discriminant)) / (2 * a);
printf("Root 1 = %.2lf and Root 2 = %.2lf\n", root1, root2);
    }
else if (discriminant == 0) {
        // One real root (repeated)
        root1 = -b / (2 * a);
        printf("Root 1 = Root 2 = %.2lf\n", root1);
    }
else
{
```

```
double realPart = -b / (2 * a);

double imaginaryPart = sqrt(-discriminant) / (2 * a);

printf("Root 1 = %.2lf + %.2lfi and Root 2 = %.2lf - %.2lfi\n",
realPart, imaginaryPart, realPart, imaginaryPart);

}

return 0;

}
```

SOURCE CODE:

c programming > C assignmentq2.c > main()

```c
#include <stdio.h>
#include <math.h>
int main() {
double a, b, c, discriminant, root1, root2;
// Input coefficients from the user
printf("Enter coefficients of the quadratic equation (a, b, c): ");
scanf("%lf %lf %lf", &a, &b, &c);
// Calculate the discriminant
discriminant = b * b - 4 * a * c;
// Check the discriminant to determine the nature of the roots
if (discriminant > 0) {
// Two real and distinct roots
root1 = (-b + sqrt(discriminant)) / (2 * a);
root2 = (-b - sqrt(discriminant)) / (2 * a);
printf("Root 1 = %.2lf and Root 2 = %.2lf\n", root1, root2);
    }
else if (discriminant == 0)
{
        // One real root (repeated)
        root1 = -b / (2 * a);
        printf("Root 1 = Root 2 = %.2lf\n", root1);
    }
else {

double realPart = -b / (2 * a);
double imaginaryPart = sqrt(-discriminant) / (2 * a);
printf("Root 1 = %.2lf + %.2lfi and Root 2 = %.2lf - %.2lfi\n", realPart, imaginaryPart, realPart, imaginaryPart);
}
return 0;
}
```

OUTPUT:

```
ssignmentq2 } ; if ($?) { .\assignmentq2 }
Enter coefficients of the quadratic equation (a, b, c): 5
5
5
Root 1 = -0.50 + 0.87i and Root 2 = -0.50 - 0.87i
PS C:\Users\user\OneDrive\Desktop\c programming> cd "c:\Users\user\OneDrive\Desktop\c programmir
ssignmentq2 } ; if ($?) { .\assignmentq2 }
Enter coefficients of the quadratic equation (a, b, c): 6
5
6
Root 1 = -0.42 + 0.91i and Root 2 = -0.42 - 0.91i
```

Q4-WAP TO CHECK HUMAN CHARACTER IS A VOWEL OR CONSONANT.

→**PROGRAM:**

```
#include <stdio.h>

int main() {

char character;

printf("Enter a character: ");

scanf("%c", &character);

// Check if the character is an alphabet (a letter)

if ((character >= 'a' && character <= 'z') || (character >= 'A' &&
character <= 'Z')) {

// Convert the character to lowercase (if it's uppercase) for easier
comparison

character = to lower(character);

// Check if the character is a vowel

if (character == 'a' || character == 'e' || character == 'i' ||
character == 'o' || character == 'u') {

 printf("%c is a vowel.\n", character);

 }

else {

            printf("%c is a consonant.\n", character);

   }

   }

else {

           printf("It's not an alphabet character.\n");

    }

     return 0;

}
```

SOURCE CODE:

c programming > C assignmentq3.c > main()

```c
#include <stdio.h>

int main() {
    char character;

    printf("Enter a character: ");
    scanf("%c", &character);

    // Check if the character is an alphabet (a letter)
    if ((character >= 'a' && character <= 'z') || (character >= 'A' && character <= 'Z')) {
        // Convert the character to lowercase (if it's uppercase) for easier comparison
        character = tolower(character);
       // Check if the character is a vowel
        if (character == 'a' || character == 'e' || character == 'i' || character == 'o' || character == 'u') {
            printf("%c is a vowel.\n", character);
        }
        else {
            printf("%c is a consonant.\n", character);
        }
    }
     else {
        // If the input is not a letter
        printf("It's not an alphabet character.\n");
    }

    return 0;
}

```

OUTPUT:

```
                    ^~~~~~~~
Enter a character: r
r is a consonant.
PS C:\Users\user\OneDrive\Desktop\c programming> cd "c:\Users\user\OneDrive\Desktop\c programming\" ; 
ssignmentq3 } ; if ($?) { .\assignmentq3 }
assignmentq3.c: In function 'main':
assignmentq3.c:12:21: warning: implicit declaration of function 'tolower' [-Wimplicit-function-declara
        character = tolower(character);
                    ^~~~~~~~
Enter a character: t
t is a consonant.
```

Q5.WAP TO FIND THE LARGEST AND SECOND LARGEST NO FROM THE GIVEN 20 NO.

→**PROGRAM:**

```
#include <stdio.h>
int main()
int n, largest, secondLargest;
// Initialize the largest and second-largest variables
largest = secondLargest = INT_MIN;
// Input 20 numbers from the user
    printf("Enter 20 numbers:\n");
    for (int i = 1; i <= 20; i++) {
        printf("Enter number %d: ", i);
        scanf("%d", &n);
// Update largest and second-largest values
  if (n > largest) {
            secondLargest = largest;
            largest = n;
        }
 else if (n > secondLargest && n != largest) {
            secondLargest = n;
        }
    }
// Display the largest and second-largest numbers
    printf("Largest number: %d\n", largest);
    printf("Second largest number: %d\n", secondLargest);
 return 0;
}
```

SOURCE CODE:

c programming > C assignmentq4.c > ...

```c
#include <stdio.h>
#include <limits.h>
int main() {
    int n, largest, secondLargest;

    // Initialize the largest and second-largest variables
    largest = secondLargest = INT_MIN; // Initialize to smallest possible integer value

    // Input 20 numbers from the user
    printf("Enter 20 numbers:\n");
    for (int i = 1; i <= 20; i++) {
        printf("Enter number %d: ", i);
        scanf("%d", &n);

        // Update largest and second-largest values
        if (n > largest) {
            secondLargest = largest;
            largest = n;
        }
        else if (n > secondLargest && n != largest) {
            secondLargest = n;
        }
    }

    // Display the largest and second-largest numbers
    printf("Largest number: %d\n", largest);
    printf("Second largest number: %d\n", secondLargest);

    return 0;
}
```

OUTPUT:

```
ssignmentq4 } ; if ($?) { .\assignmentq4 }
Enter 20 numbers:
Enter number 1: 3
Enter number 2: 4
Enter number 3: 5
Enter number 4: 6
Enter number 5: 7
Enter number 6: 8
Enter number 7: 9
Enter number 8: 11
Enter number 9: 13
Enter number 10: 12
Enter number 11: 14
Enter number 12: 15
Enter number 13: 16
Enter number 14: 17
Enter number 15: 18
Enter number 16: 19
Enter number 17: 20
Enter number 18: 21
Enter number 19: 22
Enter number 20: 23
Largest number: 23
Second largest number: 22
```