Assignment 5

Shri G.S Institute of Technology & Science

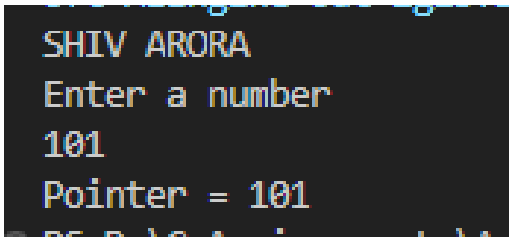
C Programming Lab

Assignment 5 – INDEX

P1. Write a program in C to show the basic declaration of a pointer.

```
#include<stdio.h>
#include<stdlib.h>
void main() {
   printf("SHIV ARORA\n");
   int *ptr;
   ptr = (int *)malloc(sizeof(int));
   printf("Enter a number\n");
   scanf("%d", ptr);
   printf("Pointer = %d\n", *ptr);
   free(ptr);
}
```

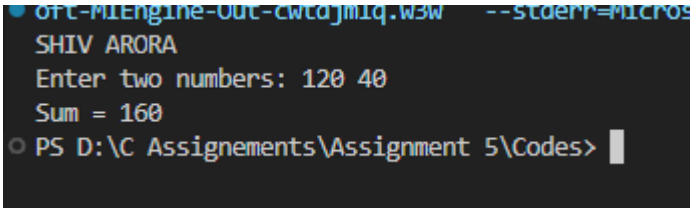
OUTPUT:



P2. Write a program in C to add two numbers using pointers.

```
#include <stdio.h>
void main() {
   printf("SHIV ARORA\n");
   int n1, n2, sum;
   int *ptr1, *ptr2, *ptr_sum;
   printf("Enter two numbers: ");
   scanf("%d %d", &n1, &n2);
   ptr1 = &n1, ptr2 = &n2, ptr_sum = &sum;
   *ptr_sum = *ptr1 + *ptr2;
   printf("Sum = %d\n", *ptr_sum);
}
```

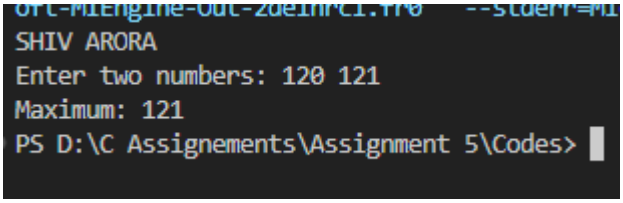
OUTPUT:



P3. Write a program in C to find the maximum number between two numbers using a pointer.

```
#include <stdio.h>
void main() {
   printf("SHIV ARORA\n");
   int n1, n2;
   int *ptr1, *ptr2;
   printf("Enter two numbers: ");
   scanf("%d %d", &n1, &n2);
   ptr1 = &n1;
   ptr2 = &n2;
   if (*ptr1 > *ptr2) printf("Maximum: %d\n", *ptr1);
   else printf("Maximum: %d\n", *ptr2);
}
```

OUTPUT:



P4. Write a program in C to store n elements in an array and print the elements using a pointer.

```
#include <stdio.h>
void main() {
printf("SHIV ARORA\n");
int size;
 printf("Enter the number of elements: ");
scanf("%d", &size);
```

```
 int arr[size];
int *ptr = arr;
 printf("Enter elements:\n");
for (int i = 0; i < size; i++) {
        scanf("%d", ptr + i);
}
printf("Elements in array are:\n");
for (int i = 0; i < size; i++) {
        printf("%d ", *(ptr + i));
}
}
```

OUTPUT:

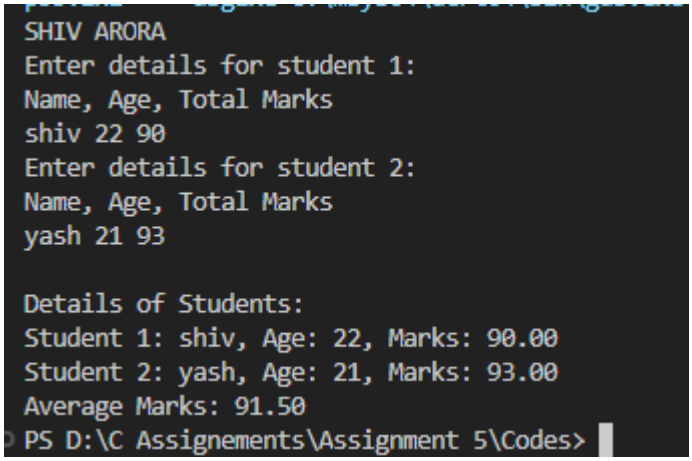
```
SHIV ARORA
Enter the number of elements: 5
Enter elements:
21 22 23 24 25
Elements in array are:
21 22 23 24 25
PS D:\C Assignements\Assignment 5\Codes>
```

P5. Create a structure called "Student" with members name, age, and total marks. Write a C program to input data for two students, display their information, and find the average of total marks.

```
#include <stdio.h>
typedef struct Student {
   char name[50];
   int age;
   float total_marks;
}Student;
void main() {
   printf("SHIV ARORA\n");
   Student s1, s2;
   printf("Enter details for student 1:\n");
   printf("Name, Age, Total Marks \n");
   scanf("%s %d %f", s1.name, &s1.age, &s1.total_marks);
```

```
printf("Enter details for student 2:\n");
printf("Name, Age, Total Marks \n");
scanf("%s %d %f", s2.name, &s2.age, &s2.total_marks);
printf("\nDetails of Students:\n");
printf("Student 1: %s, Age: %d, Marks: %.2f\n", s1.name, s1.age, s1.total_marks);
printf("Student 2: %s, Age: %d, Marks: %.2f\n", s2.name, s2.age, s2.total_marks);
float average = (s1.total_marks + s2.total_marks) / 2;
printf("Average Marks: %.2f\n", average);
}
```

OUTPUT:

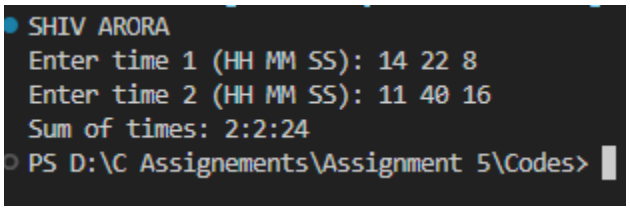


P6. Define a structure named Time with members hours, minutes, and seconds. Write a C program to input two times, add them, and display the result in proper time format.

```
#include <stdio.h>
typedef struct Time {
    int hours;
    int minutes;
    int seconds;
} Time;
Time addTimes(Time t1, Time t2) {
    Time result;
    result.seconds = t1.seconds + t2.seconds;
    result.minutes = t1.minutes + t2.minutes + result.seconds / 60;
    result.seconds %= 60;
```

```
    result.hours = t1.hours + t2.hours + result.minutes / 60;
    result.minutes %= 60;
    result.hours %= 24;
    return result;
}
void main() {
    printf("SHIV ARORA\n");
    Time t1, t2, result;
    printf("Enter time 1 (HH MM SS): ");
    scanf("%d %d %d", &t1.hours, &t1.minutes, &t1.seconds);
    printf("Enter time 2 (HH MM SS): ");
    scanf("%d %d %d", &t2.hours, &t2.minutes, &t2.seconds);
    result = addTimes(t1, t2);
    printf("Sum of times: %d:%d:%d\n", result.hours, result.minutes, result.seconds);
}
```

OUTPUT:

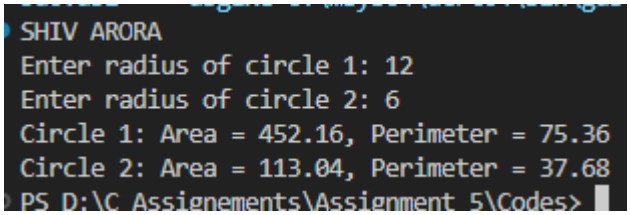


P7. Define a structure named Circle to represent a circle with a radius. Write a C program to calculate the area and perimeter of two circles and display the results.

```
#include <stdio.h>
#include <math.h>
#define M_PI 3.14
typedef struct Circle {
    float radius;
}Circle;
void calculate(Circle c, float *area, float *perimeter) {
    *area = M_PI * c.radius * c.radius;
    *perimeter = 2 * M_PI * c.radius;
}
```

```
void main() {
    printf("SHIV ARORA\n");
    Circle c1, c2;
    float area1, perimeter1, area2, perimeter2;
    printf("Enter radius of circle 1: ");
    scanf("%f", &c1.radius);
    printf("Enter radius of circle 2: ");
    scanf("%f", &c2.radius);
    calculate(c1, &area1, &perimeter1);
    calculate(c2, &area2, &perimeter2);
    printf("Circle 1: Area = %.2f, Perimeter = %.2f\n", area1, perimeter1);
    printf("Circle 2: Area = %.2f, Perimeter = %.2f\n", area2, perimeter2);
}
```

OUTPUT:

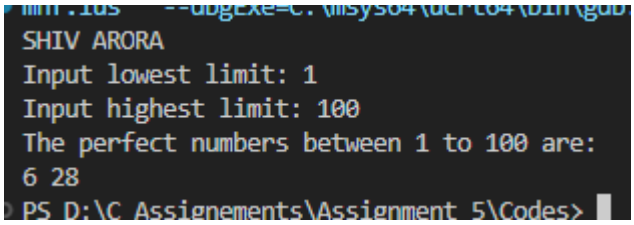


P8. Write a program in C to print all perfect numbers in a given range using the function.

```
#include <stdio.h>
#include<stdbool.h>
bool isPerfect(int n) {
    int sum = 0;
    for (int i = 1; i <= n / 2; i++) {
        if (n % i == 0) {
            sum += i;
        }
    }
    return sum == n;
}
void findPerfectNumbers(int low, int high) {
```

```
    printf("The perfect numbers between %d to %d are:\n", low, high);
    for (int i = low; i <= high; i++) {
        if (isPerfect(i)) {
            printf("%d ", i);
        }
    }
    printf("\n");
}
void main() {
    printf("SHIV ARORA\n");
    int low, high;
    printf("Input lowest limit: ");
    scanf("%d", &low);
    printf("Input highest limit: ");
    scanf("%d", &high);
    findPerfectNumbers(low, high);
}
```

OUTPUT:

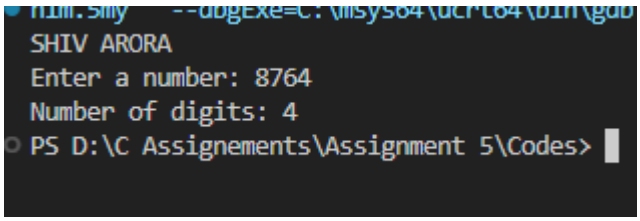


P9. Write a program in C to count the digits of a given number using recursion.

```
#include <stdio.h>
int countDigits(int n) {
    if (n == 0) {
        return 0;
    }
    return 1 + countDigits(n / 10);
}
void main() {
```

```
    printf("SHIV ARORA\n");
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    printf("Number of digits: %d\n", countDigits(n));
}
```

OUTPUT:

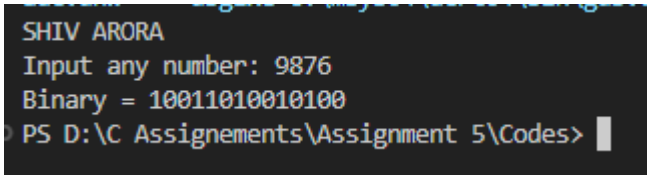


P10. Write a program in C to convert a decimal number to binary using recursion.

```
#include <stdio.h>
void decimalToBinary(int num) {
    if (num == 0) {
        return;
    }
    decimalToBinary(num / 2);
    printf("%d", num % 2);
}
void main() {
    printf("SHIV ARORA\n");
    int num;
    printf("Input any number: ");
    scanf("%d", &num);
    printf("Binary = ");
    if (num == 0) {
        printf("0");
    } else {
        decimalToBinary(num);
    }
```

```
    printf("\n");
}
```

OUTPUT:

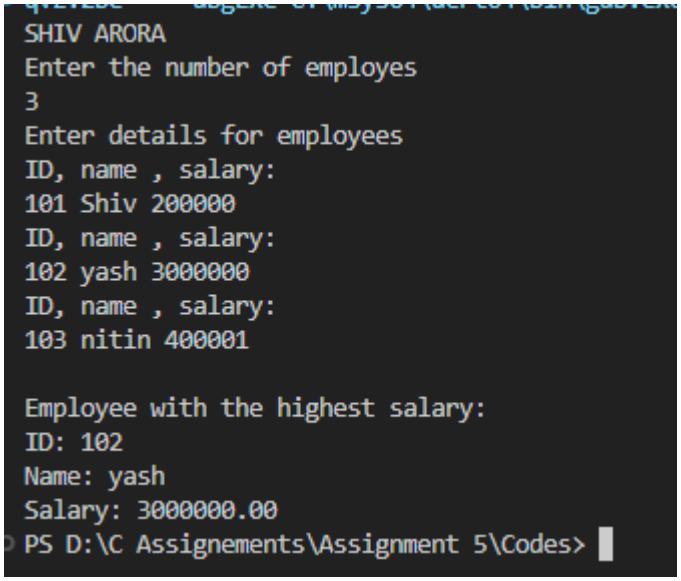


P11. Create a structure named "Employee" to store employee details such as employee ID, name, and salary. Write a program to input data for three employees, find the highest salary employee, and display their information.

```
#include <stdio.h>
#include <string.h>
typedef struct Employee {
    int id;
    char name[50];
    float salary;
}Employee;
void main() {
    printf("SHIV ARORA\n");
    int size;
    printf("Enter the number of employes \n");
    scanf("%d", &size);
    Employee emp[size];
    int highestIndex = 0;
    printf("Enter details for employees\n");
    for (int i = 0; i < size; i++) {
        printf("ID, name , salary: \n");
        scanf("%d %s %f", &emp[i].id, emp[i].name, &emp[i].salary);
        if (emp[i].salary > emp[highestIndex].salary) highestIndex = i;
    }
    printf("\nEmployee with the highest salary:\n");
    printf("ID: %d\n", emp[highestIndex].id);
```

```
        printf("Name: %s\n", emp[highestIndex].name);
        printf("Salary: %.2f\n", emp[highestIndex].salary);
    }
```

OUTPUT:



P12. Write a program in C to find the largest element using Dynamic Memory Allocation.

```
#include <stdio.h>
#include <stdlib.h>
void main() {
    printf("SHIV ARORA\n");
    int size;
    int *arr, largest;
    printf("Enter the number of elements: ");
    scanf("%d", &size);
    arr = (int *)malloc(size * sizeof(int));
    printf("Enter elements:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    largest = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > largest) {
```

```
                largest = arr[i];
            }
        }
        printf("Largest = %d\n", largest);
        free(arr);
    }
```

OUTPUT: