PRACTICAL – 6

PROGRAM -1

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

CODE:-

|  |
| --- |
| #include <stdio.h>  int main() {  printf("\n HARSH D \n");  int num = 10;  int \*ptr;  ptr = &num;  printf("Value of num: %d\n", num);  printf("Address of num: %p\n", &num);  printf("Value of num using pointer: %d\n", \*ptr);  printf("Address of num using pointer: %p\n", ptr);  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -2

AIM- Write a program in C to demonstrate the use of the & (address of) and \*(value at address) operators.

CODE:-

|  |
| --- |
| #include <stdio.h>  int main() {  printf("\n HARSH D \n");  int num = 10;  int \*ptr;  ptr = &num;  printf("Value of num: %d\n", num);  printf("Address of num: %p\n", &num);  printf("Value of num using pointer: %d\n", \*ptr);  printf("Address of num using pointer: %p\n", ptr);  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -3

AIM- Write a program in C to add numbers using call by reference.

CODE:-

|  |
| --- |
| #include <stdio.h>  void addNumbers(int \*a, int \*b, int \*sum) {  \*sum = \*a + \*b;  }  int main()  {  printf("\n HARSH D \n");  int num1, num2, sum;  printf("Enter first number: ");  scanf("%d", &num1);  printf("Enter second number: ");  scanf("%d", &num2);  addNumbers(&num1, &num2, &sum);  printf("Sum of %d and %d is: %d\n", num1, num2, sum);  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -4

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

CODE:-

|  |
| --- |
| #include <stdio.h>  int main()  {  printf("\n HARSH D \n");  int n;  printf("Enter the number of elements: ");  scanf("%d", &n);  int arr[n];  printf("Enter %d elements:\n", n);  for (int i = 0; i < n; i++) {  scanf("%d", &arr[i]);  }  printf("You entered:\n");  int \*ptr = arr;  for (int i = 0; i < n; i++) {  printf("%d ", \*(ptr + i));  }  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -5

AIM- Write a program in C to sort an array using a pointer.

CODE:-

|  |
| --- |
| #include <stdio.h>  void sortArray(int \*arr, int size) {  int i, j, temp;  // Bubble sort  for (i = 0; i < size - 1; i++) {  for (j = 0; j < size - i - 1; j++) {  if (\*(arr + j) > \*(arr + j + 1)) {  // Swap the elements  temp = \*(arr + j);  \*(arr + j) = \*(arr + j + 1);  \*(arr + j + 1) = temp;  }  }  }  }  int main()  {  printf("\n HARSH D \n");  int n;  printf("Enter the number of elements: ");  scanf("%d", &n);  int arr[n];  printf("Enter %d elements:\n", n);  for (int i = 0; i < n; i++) {  scanf("%d", &arr[i]);  }  sortArray(arr, n);  printf("Sorted array:\n");  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -6

AIM- Write a program in C to demonstrate the use of pointers to structures. CODE:-

|  |
| --- |
| #include <stdio.h>  #include <string.h>  struct Student {  int id;  char name[50];  float percentage;  };  int main()  {  printf("\n HARSH D \n");  struct Student student1;  struct Student \*ptrStudent;  ptrStudent = &student1;  // Assign values using pointer  ptrStudent->id = 1;  strcpy(ptrStudent->name, "John");  ptrStudent->percentage = 85.5;  // Access values using pointer  printf("Student ID: %d\n", ptrStudent->id);  printf("Student Name: %s\n", ptrStudent->name);  printf("Student Percentage: %.2f\n", ptrStudent->percentage);  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |