**PF Lab No 9**

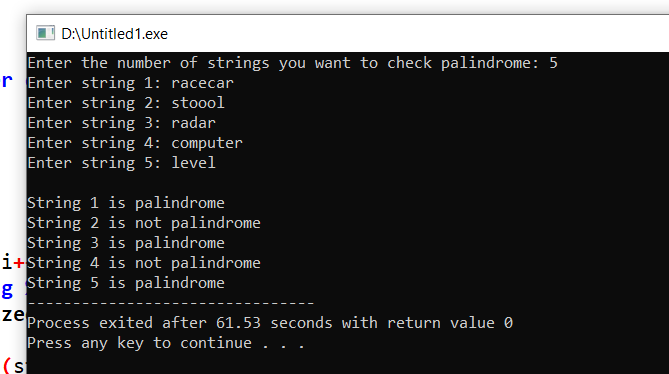
**Name: Ali Rooman**

**Roll No: 24K-0792**

1. Create a program that accepts a 2D array of strings (e.g., 5 words with a max length of 20 characters each). Determines if each word (row) is a palindrome. Outputs “Palindrome” or “Not Palindrome” for each word. A palindrome is a word that reads the same forward and backward.

|  |
| --- |
| #include <stdio.h>  #include <string.h>  void palindrome(char string[][20], int n) {      char temp[100][100];      int i, j;        for (i = 0; i < n; i++) {          int length = strlen(string[i]);          for (j = 0; j < length; j++) {              temp[i][j] = string[i][length - 1 - j];          }          temp[i][length] = '\0';      }        for (i = 0; i < n; i++) {          if (strcmp(string[i], temp[i]) == 0) {              printf("\nString %d is palindrome", i + 1);          } else {              printf("\nString %d is not palindrome", i + 1);          }      }  }  int main() {      printf("Enter the number of strings you want to check palindrome: ");      int n,i;      scanf("%d", &n);      getchar();      char string[n][20];        for (i = 0; i < n; i++) {          printf("Enter string %d: ", i + 1);          fgets(string[i], sizeof(string[i]), stdin);            size\_t len = strlen(string[i]);          if (len > 0 && string[i][len - 1] == '\n') {              string[i][len - 1] = '\0';          }      }      palindrome(string, n);        return 0;  } |

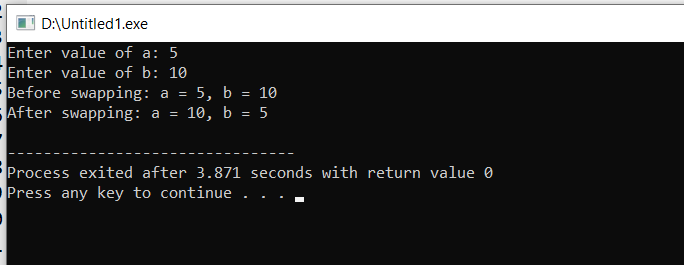
OUTPUT



2. Create a C program that swaps the values of two integers using a user-defined function, swapIntegers. The user inputs two integer values, and the program uses the function to swap them. It should perform the swap and display the updated values.

|  |
| --- |
| #include <stdio.h>  void swap(int a, int b) {  printf("Before swapping: a = %d, b = %d\n", a, b);  a = a + b;  b = a - b;  a = a - b;  printf("After swapping: a = %d, b = %d\n", a, b);  }  int main() {  int a,b;  printf("Enter value of a: ");  scanf("%d",&a);  printf("Enter value of b: ");  scanf("%d",&b);  swap(a, b);  return 0;  } |

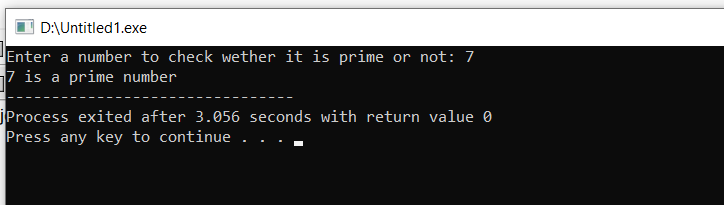
OUTPUT



3. Implement a function that checks if a given integer is a prime number. Use this function in the main program to check if numbers entered by the user are prime.

|  |
| --- |
| #include <stdio.h>  #include <math.h>  void isPrime(int n){  int prime = 1,i;  if(n<2){  printf("The number is not a prime number");  }  else{  for(i=2;i<sqrt(n);i++){  if(n%i==0){  prime=0;  break;  }  }  }  if(prime){  printf("%d is a prime number",n);  }  else{  printf("%d is not a prime number",n);  }  }  int main() {  int a;  printf("Enter a number to check whether it is prime or not: ");  scanf("%d",&a);  isPrime(a);  return 0;  } |

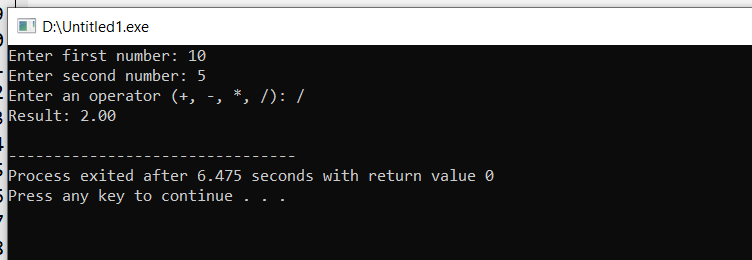
OUTPUT



4. Write a C program with a user-defined function calculate to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. The program should take two numbers and an operation choice as input, and then use the function to perform the operation.

|  |
| --- |
| #include <stdio.h>  float calculate(float num1, float num2, char operator) {  float result;    switch (operator) {  case '+':  result = num1 + num2;  break;  case '-':  result = num1 - num2;  break;  case '\*':  result = num1 \* num2;  break;  case '/':  if (num2 != 0) {  result = num1 / num2;  } else {  printf("Error: Division by zero is not allowed.\n");  result = 0;  }  break;  default:  printf("Error: Invalid operator.\n");  result = 0;  break;  }    return result;  }  int main() {  float num1, num2, result;  char operator;    printf("Enter first number: ");  scanf("%f", &num1);  printf("Enter second number: ");  scanf("%f", &num2);  printf("Enter an operator (+, -, \*, /): ");  scanf(" %c", &operator);  result = calculate(num1, num2, operator);  printf("Result: %.2f\n", result);  return 0;  } |

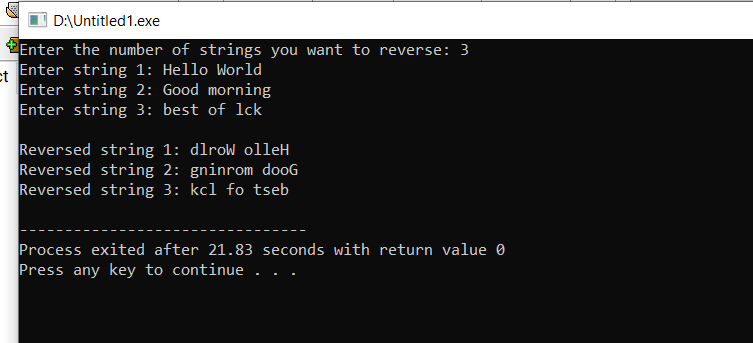
OUTPUT



5. Create a function that reverses a given string and returns the reversed string. Use this function in the main program to display the reversed string entered by the user.

|  |
| --- |
| #include <stdio.h>  #include <string.h>  void rev(char string[][20], int n) {  char temp[100][100];  for (int i = 0; i < n; i++) {  int length = strlen(string[i]);  for (int j = 0; j < length; j++) {  temp[i][j] = string[i][length - 1 - j];  }  temp[i][length] = '\0';  }  printf("\n");  for (int i = 0; i < n; i++) {  printf("Reversed string %d: %s\n",i+1, temp[i]);  }  }  int main() {  printf("Enter the number of strings you want to reverse: ");  int n;  scanf("%d", &n);  getchar();  char string[n][20];    for (int i = 0; i < n; i++) {  printf("Enter string %d: ", i + 1);  fgets(string[i], sizeof(string[i]), stdin);    size\_t len = strlen(string[i]);  if (len > 0 && string[i][len - 1] == '\n') {  string[i][len - 1] = '\0';  }  }  rev(string, n);    return 0;  } |

OUTPUT



6. Create a function that returns the maximum and minimum element in an integer array. Use this function in the main program to find the maximum and minimum from an array entered by the user.

|  |
| --- |
| #include <stdio.h>  int arrayment(int arr[], int n) {      int min = arr[0];      int i;      for (i = 1; i < n; i++) {          if (arr[i] < min) {              min = arr[i];          }      }      return min;  }  int arraymax(int arr[], int n) {      int max = arr[0];      int i;      for (i = 1; i < n; i++) {          if (arr[i] > max) {              max = arr[i];          }      }      return max;  }  int main() {      int n, i;      printf("Enter size of array: ");      scanf("%d", &n);      int vec[n];      printf("Enter %d elements in array: \n", n);      for (i = 0; i < n; i++) {          printf("Element %d: ", i + 1);          scanf("%d", &vec[i]);      }      printf("Maximum element: %d\n", arraymax(vec, n));      printf("Minimum element: %d\n", arraymin(vec, n));      return 0;  } |

OUTPUT

