Day – 7

41. Write a c program to find palindrome no using string.

Ans: Input:

#include <stdio.h>

#include <string.h>

int main() {

char num[100], rev[100];

printf("Enter a number: ");

scanf("%s", num);

strcpy(rev, num);

strrev(rev);

if (strcmp(num, rev) == 0)

printf("The number is a palindrome.\n");

} else {

printf("The number is not a palindrome.\n");

}

return 0;

}

Output:

Enter a number: 121

The number is a palindrome.

42. Write a c program to check whether given string is a sub-string of an existing string.

Ans: Input:

#include <stdio.h>

#include <string.h>

int main() {

char str[100], sub[50];

printf("Enter the main string: ");

gets(str);

printf("Enter the substring: ");

gets(sub);

if (strstr(str, sub)!=NULL) {

printf("'%s' is a substring of '%s'\n", sub, str);

} else {

printf("'%s' is not a substring of '%s'\n", sub, str);

}

return 0;

}

Output:

Enter the main string : programming in c

Enter the substring : in

In is a substring of programming in c

43. Write a c program to calculate no of vowels, consonants, white space and num in a string.

Ans: Input:

#include <stdio.h>

Int main() {

Char str[200];

Int vowels = 0, consonants = 0, spaces = 0, digits = 0;

Printf(“Enter a string: “);

gets(str);

For (int i = 0; str[i] != ‘\0’; i++) {

Char ch = str[i];

If (ch >= ‘a’ && ch <= ‘z’) {

If (ch == ‘a’ || ch == ‘e’ || ch == ‘i’ || ch == ‘o’ || ch == ‘u’ || ch == ‘A’ || ch == ‘E’ || ch == ‘I’ || ch == ‘O’ || ch == ‘U’)

Vowels++;

Else

Consonants++;

} else if (ch >= ‘0’ && ch <= ‘9’) {

Digits++;

} else if (ch == ‘ ‘) {

Spaces++;

}

}

Printf(“Vowels: %d\n”, vowels);

Printf(“Consonants: %d\n”, consonants);

Printf(“White spaces: %d\n”, spaces);

Printf(“Numbers: %d\n”, digits);

Return 0;

}

Output:

Enter a string: Hello World 123

Vowels: 3

Consonants: 7

White spaces: 2

Digits: 3

44. Write a c program to find ASCII value of a string.

Ans: Input:

#include <stdio.h>

Int main() {

Char str[100];

Printf(“Enter a string: “);

Scanf(“ %[^\n]”, str);

Printf(“ASCII values:\n”);

For (int i = 0; str[i] != ‘\0’; i++) {

Printf(“’%c’ = %d\n”, str[i], str[i]);

}

Return 0;

}

Output:

Enter a string: Hi!

ASCII values:

‘H’ = 72

‘i’ = 105

‘!’ = 33

45. Write a c program to find the length of a string and print the string in reverse order without using library function.

Ans: Input:

#include <stdio.h>

Int main() {

Char str[100];

Int length = 0;

Printf(“Enter a string: “);

Scanf(“ %[^\n]”, str);

While (str[length] != ‘\0’) {

Length++;

}

Printf(“Length of the string: %d\n”, length);

Printf(“Reversed string: “);

For (int i = length – 1; i >= 0; i--) {

Printf(“%c”, str[i]);

}

Printf(“\n”);

Return 0;

}

Output:

Enter a string: Hello World

Length of the string: 11

Reversed string: dlroW olleH

46. Write a c program to find bubble sort.

Ans: Input:

#include <stdio.h>

Int main() {

Int a[100], n, i, j, t;

Printf(“Enter number of elements: “);

Scanf(“%d”, &n);

Printf(“Enter %d elements:\n”, n);

For (i = 0; i < n; i++) {

Scanf(“%d”, &a[i]);

}

For (i = 0; i < n – 1; i++) {

For (j = 0; j < n – i – 1; j++) {

If (a[j] > a[j + 1]) {

t = a[j];

A[j] = a[j + 1];

A[j + 1] = t;

}

}

}

Printf(“Sorted array in ascending order:\n”);

For (i = 0; i < n; i++) {

Printf(“%d “, a[i]);

}

Printf(“\n”);

Return 0;

}

Output:

Enter number of elements: 5

Enter 5 elements:

4 2 5 1 3

Sorted array in ascending order:

1 2 3 4 5

47. Write a c program to find recursion binary search of a no.

Ans: Input:

#include <stdio.h>

Int binarySearch(int arr[], int low, int high, int x) {

If (low > high)

Return -1;

Int mid = (low + high) / 2;

If (arr[mid] == x)

Return mid;

Else if (x < arr[mid])

Return binarySearch(arr, low, mid – 1, x);

Else

Return binarySearch(arr, mid + 1, high, x);

}

Int main() {

Int arr[100], n, x, i, result;

Printf(“Enter number of elements: “);

Scanf(“%d”, &n);

Printf(“Enter %d sorted elements:\n”, n);

For (i = 0; i < n; i++) {

Scanf(“%d”, &arr[i]);

}

Printf(“Enter the number to search:”);

Scanf(“%d”, &x);

Result = binarySearch(arr, 0, n – 1, x);

If (result != -1)

Printf(“Element found at index %d\n”, result);

Else

Printf(“Element not found\n”);

Return 0;

}

Output:

Enter number of elements: 5

Enter 5 sorted elements:

1 3 5 7 9

Enter the number to search: 5

Element found at index 2