Q) Program to find number of vowels, consonants, spaces, other characters in a string

**PROGRAM CODE**

#include <stdio.h>

#include <stdlib.h>

void main()

{

char str[200];

int i, vowels=0,consonants=0,digits=0,spaces=0,specialCharacters=0;

printf("Enter a string: ");

gets(str);

for (i = 0; str[i] != '\0'; i++)

{

if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' || str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U')

{

vowels++;

}

else if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))

{

consonants++;

}

else if (str[i] >= '0' && str[i] <= '9')

{

digits++;

}

else if (str[i] == ' ')

{

spaces++;

}

else

{

specialCharacters++;

}

}

printf("\nVowels = %d", vowels);

printf("\nConsonants = %d", consonants);

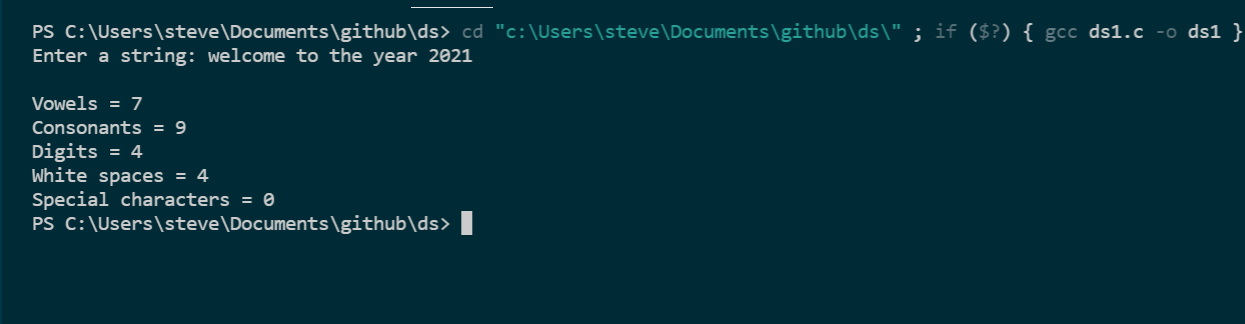
printf("\nDigits = %d", digits);

printf("\nWhite spaces = %d", spaces);

printf("\nSpecial characters = %d", specialCharacters);

}

**OUTPUT**



Q) Program to search for a substring in a string.

**PROGRAM CODE**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i, j, temp;

char str[100] = {"This is a pattern matching"};

char substr[20] = {"pattern"};

for (i = 0; str[i] != '\0'; i++)

{

j = 0;

if (str[i] == substr[j])

{

temp = i + 1;

while (str[i] == substr[j])

{

i++;

j++;

}

if (substr[j] == '\0')

{

printf("The substring is present in given string at position %d\n", temp);

exit(0);

}

else

{

i = temp;

temp = 0;

}

}

}

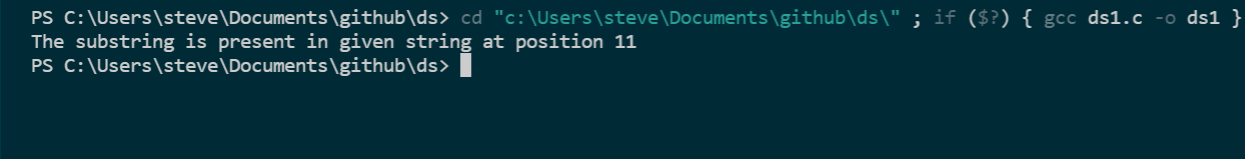
if (temp == 0)

printf("The substring is not present in given string\n");

return 0;

}

**OUTPUT**



Q) Program to check whether a given string is palindrome or not

**PROGRAM CODE**

#include <stdio.h>

#include <string.h>

void isPalindrome(char str[])

{

int l = 0;

int h = strlen(str) - 1;

while (h > l)

{

if (str[l++] != str[h--])

{

printf("%s is Not Palindrome", str);

return;

}

}

printf("%s is palindrome\n", str);

}

int main()

{

char str[50];

printf("Enter the string :");

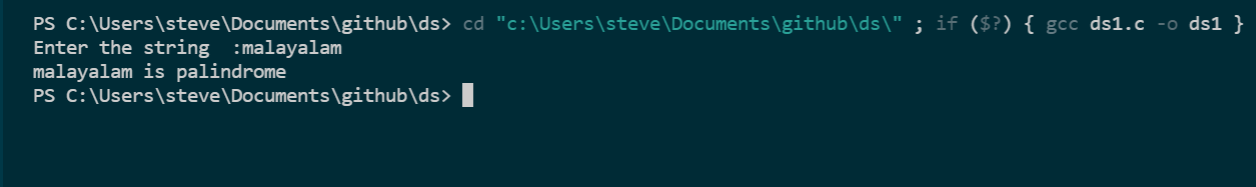
gets(str);

isPalindrome(str);

return 0;

}

**OUTPUT**



**Program to represent Searching procedures (Linear search , Binary search Interpolation search)**

Q)Linear search

**PROGRAM CODE**

#include <stdio.h>

void main()

{

int a[10], item, i, n;

printf("Enter the no of elements");

scanf("%d", &n);

printf("\nEnter the elements");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

printf("\nEnter the element to be searched");

scanf("%d", &item);

for (i = 0; i < n; i++)

{

if (a[i] == item)

{

printf("\nItem found at index %d", i + 1);

break;

}

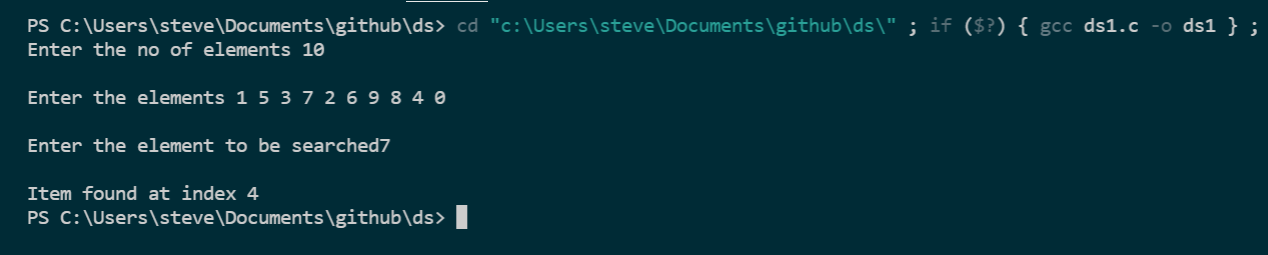
}

if (i >= n)

printf("Item not found");

}

**OUTPUT**



Q) Binary search

**PROGRAM CODE**

#include <stdio.h>

void main()

{

int c, first, last, middle, n, search, array[100];

printf("Enter number of elements\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

for (c = 0; c < n; c++)

scanf("%d", &array[c]);

printf("Enter value to find\n");

scanf("%d", &search);

first = 0;

last = n - 1;

middle = (first + last) / 2;

while (first <= last)

{

if (array[middle] < search)

first = middle + 1;

else if (array[middle] == search)

{

printf("%d found at location %d.\n", search, middle + 1);

break;

}

else

last = middle - 1;

middle = (first + last) / 2;

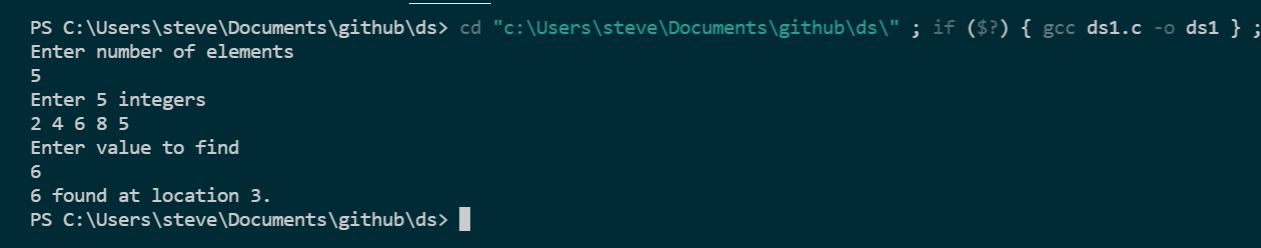
}

if (first > last)

printf("Not found! %d isn't present in the list.\n", search);

}

**OUTPUT**



Q) Interpolation search

**PROGRAM CODE**

#include <stdio.h>

void main()

{

int n, a[10], item, i, first = 0, last, mid, flag = 0;

printf("\nEnter the number of elements : ");

scanf("%d", &n);

printf("\n\nEnter the elements : ");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

last = n - 1;

printf("\nEnter the element to be searched : ");

scanf("%d", &item);

while (first <= last)

{

mid = first + ((item - a[first]) \* (last - first)) / (a[last] - a[first]);

if (a[mid] == item)

{

printf("\nThe element is found at position %d", mid + 1);

flag = 1;

break;

}

else if (a[mid] < item)

{

first = mid + 1;

}

else

{

last = mid - 1;

}

}

if (flag == 0)

{

printf("\nThe element not found...");

}

}

**OUTPUT**

