**1) Write a function to print a given string in reverse**

**CODE 1 :**

// 1.Write a function to print a given string in reverse.

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

printf("Enter the string: ");

gets(str);

int count = 0;

printf("The reverse string is: ");

for (int i = 1; str[i] != '\0'; i++)

{

count++;

}

for (int j = count; j >= 0; j--)

{

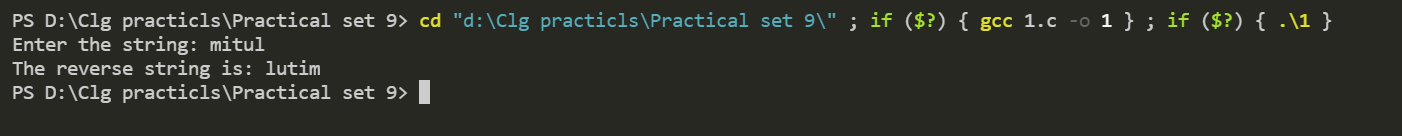
printf("%c", str[j]);

}

return 0;

}

**OUTPUT 1 :**

****

**2) Write a function strindex((s,t) to find index of string t into string s. If s=”I am using unix” and t=”am”, then function returns 3.**

**CODE 2 :**

/\* 2.Write a function strindex(s,t) to find the index of string t into string

s.If s="I am using unix" and t"am", then function returns 3.\*/

#include <stdio.h>

#include <string.h>

int strindex(char \*s, char \*t)

{

int l1, l2, diff, i;

l1 = strlen(s);

l2 = strlen(t);

diff = l1 - l2;

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

{

if (strncmp(s + i, t, l2) == 0)

{

return i;

}

}

}

int main()

{

int i, diff;

char str[100], sub[50];

printf("Enter the the main string: ");

gets(str);

printf("Enter the sub string to be search: ");

gets(sub);

diff = strlen(str) - strlen(sub);

i = strindex(str, sub);

if (i <= diff)

{

printf("String found at index: %d", i + 1);

}

else

{

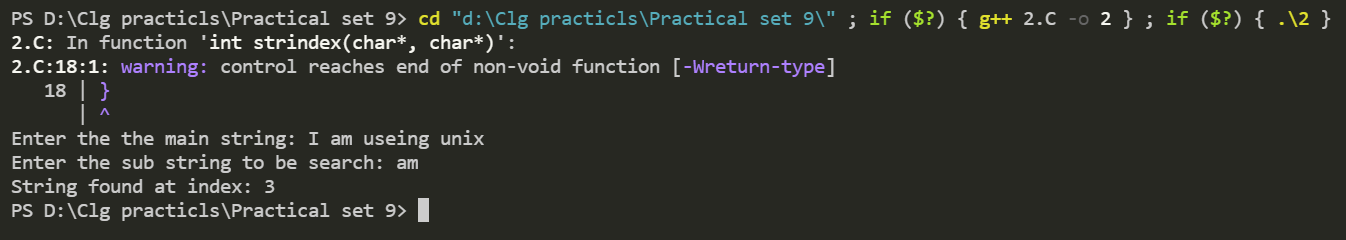
printf("There is no match.");

}

return 0;

}

**OUTPUT 2 :**

****

**3) Write a function atoi(s) to convert string of digits into a number.**

**CODE 3 :**

// 3.Write a function atoi(s) to convert a string of digits into a number.

#include <stdio.h>

#include <string.h>

#include<math.h>

int myatoi(char \*str)

{

int sum = 0;

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

{

sum = sum \* 10 + str[i] - '0';

}

return sum;

}

int main()

{

int result;

char str[20];

printf("Enter the string: ");

scanf("%s", str);

printf("String value: %s\n", str);

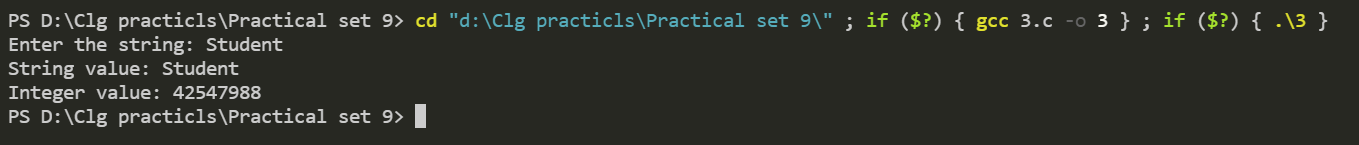
result = myatoi(str);

printf("Integer value: %d\n", result);

return 0;

}

**OUTPUT 3 :**

****

**4) Write a program to sort the integers given as command line arguments.**

**CODE 4 :**

// 4.Write a program to sort the integers given as command line arguments.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(int argc, char \*argv[])

{

int min[argc - 1], temp, len, i, j;

printf("\nThe value of argc is %d\n", argc);

printf("\n");

printf("Entered integers type arguments are: ");

for (int i = 0; i < argc - 1; i++)

{

min[i] = atoi(argv[i + 1]);

printf("%d ", min[i]);

}

printf("\n");

printf("\nSorted integers in Ascending order: ");

for (int i = 0; i < argc - 1; i++)

{

for (int j = i; j < argc - 1; j++)

{

if (min[i] > min[j])

{

temp = min[j];

min[j] = min[i];

min[i] = temp;

}

}

}

for (int i = 0; i < argc - 1; i++)

{

printf("%d ", min[i]);

}

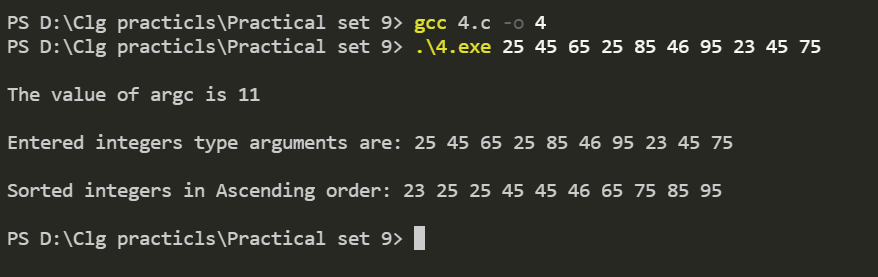
printf("\n");

printf("\n");

return 0;

}

**OUTPUT 4 :**

****

**5) Write a function to reverse a string using pointers. Use it in main() to reverse a string given from keyboard.**

**CODE 5 :**

/\* 5.Write a function to reverse a string using pointers. Use it in main() to reverse a string given from the keyboard.\*/

#include <stdio.h>

int string\_length(char \*);

void reverse(char \*);

int main()

{

char string[100];

printf("Enter a string: ");

gets(string);

reverse(string);

printf("Reverse of entered string is \"%s\".\n", string);

return 0;

}

void reverse(char \*string)

{

int length, c;

char \*begin, \*end, temp;

length = string\_length(string);

begin = string;

end = string;

for (c = 0; c < (length - 1); c++)

end++;

for (c = 0; c < length / 2; c++)

{

temp = \*end;

\*end = \*begin;

\*begin = temp;

begin++;

end--;

}

}

int string\_length(char \*pointer)

{

int c = 0;

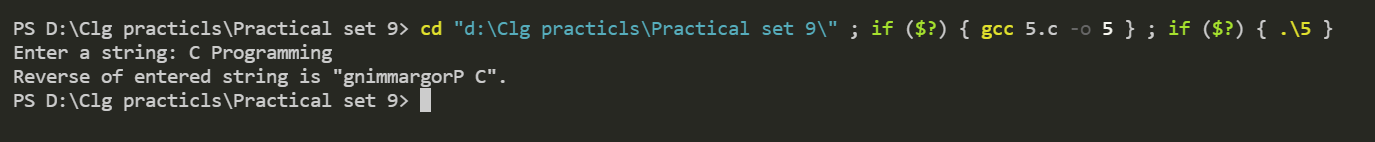
while (\*(pointer + c) != '\0')

c++;

return c;

}

**OUTPUT 5 :**

****