Enroll - NBTG13715
Name - Nitin chaudhary
Batch - F8
Tut - 10

Date 13-12-2021
Page

Ans ①→

```
#include<stdio.h>
void reverse(char a[])
{
   int i=strlen(a);
   printf("reversed string is \n");
   for(i-1; i>=0;i--)
   {
      printf("%c", a[i]);
   }
}
int main()
{
   char a[100];
   printf("enter a string \n");
   gets(a);
   reverse(a);
}
```

Output -

```
enter a string
nitinchaudhary
reversed string is
yrahduahcnitin
```

Ans (2)

```
#include<stdio.h>
int reverse(char a[])
{
 int i, check=0;
 for(i=0; i<=strlen(a)/2 ; i++)
 {
    if(a[i]==a[strlen(a)-i-1])
        check++;
 }
 if (check==i)
 {
 printf("given string is palindrome");
 return 1;
 }
 else
 {
 printf("given string is not a palindrome");
 return 0;
 }
}
int main()
{
    char a[100];
    printf("enter a string\n");
    gets(a);
    reverse(a);
}
```

Output:

enter a string
nitin
given string is palindrome

Ans(3) →

```
#include <stdio.h>
void concate(char a[], char b[])
{
 strcat (a, b);
 printf (" concatenated string is \n");
 puts (a);
}
int main(){
char a [100], b [100];
 printf(" enter first string \n");
 gets (a) ;
 printf(" enter second string \n");
 gets(b);
 concate (a, b);
}
```

Output →

```
enter first string
nitin
enter second string
chaudhary
concatenated string is
nitinchaudhary
```

Ans (9) →

```
#include <stdio.h>
#include <string.h>
void PrintArray(char* arr[], int n)
{
for(int i=0; i<n; i++) {
printf("%s", arr[i]);
 }
}
void ReverseArray(char *arr[], int n)
{
char* temp;
int j=n-1;
for(int i=0; i<j; i++){
temp=arr[i];
arr[i] = arr[j];
arr[j] = temp;
j--;
 }
}
int main()
{
char* arr[] = {"Anaand", "Naureen", "Banjot
"Wahid", "Sheena", "Ali", "Simran"};
int n= sizeof(arr) /sizeof(arr[0]);
PrintArray(arr, n);
printf("\n");
ReverseArray(arr, n);
PrintArray(arr, n);
return 0;
}
```

Output →

Anaand Naureen Banjot Wahid Sheena Ali Simran

Simran Ali Sheena Wahid Banjot Naureen Anaand

Ans (5) (a)
```
#include <stdio.h>
#include <string.h>
int len(char *p)
{ int l=0;
  while (*p != 0) {
    l++;
    p++;
  }
  printf("The length of string is: %d", l);
}
```

(b)
```
void rev(char *p)
{ char *t;
  int q = strlen(p), j = q-1;
  for (int i=0; i<q/2; i++)
  { t = p[j];
    p[j] = p[i];
    p[i] = t;
    j--;
  }
  printf("The reversed string is: ");
  puts(p);
}
```

(c)
```
void con(char *a, char *b)
{ int l1 = strlen(a), l2 = strlen(b);
  for (int i=0; i<l1+l2; i++)
  { if (i>=l1)
      { a[i] = b[i-l1]; } }
  printf("The Concatenated string is:");
  puts(a);
}
```

(d)
```
void copy (char *a, char *b)
{ int l1 = strlen(a), l2 = strlen(b), i=0;
  printf("%d", l1);
  while (i<l1)
  {
    a[i] = b[i];
    i++;
    if (i>b)
    a[i] = '\0'; }
  printf("copied string:");
  puts(a);
}
```

(e)
```
int con( char *a; char *b)
{
  int l1 = strlen(a), l2 = strlen[b), i=0;
  int s;
  if (l1>l2)
   s=l1;
  if (l2>l1)
   s=l2;
   while (i<s)
   { if(a[i] > b[i])
    return 1;
      else if (a[i] < b[i])
       return -1;
      i++;
    }
  }
int main()
{char a[20], b[20], c[20];
```

```
printf ("Enter a string");
gets (a)
len (a);
strcpy (c,a);
read a);
printf("Enter the second string:");
gets (b);
con (c,b);
copy (a,b);
printf ("Enter two strings to compare: \n");
gets (a);
gets (b);
int v = com(a,b);
print ("%d", 1);
}
```

Output →

Enter a string : nitin
The length of string is : 5
The reversed string is nitin
Enter the second string : chaudhary
The concatenated string is : nitinchaudhary
Copied string is : chaudhary
Enter two string to compare :
string
strong
-1

(6)

```
#include <stdio.h>
#define Row 3
#define CoL 3
void matrixinput (int mat [ ] [CoL])
 { int row, col;
    for (row = 0; row < Row ; row++)
    {
     for ( col = 0; col < CoL ; col++ )
       scanf ("%d", (*(mat + row) + col));
    }
 }
void matrix print (int mat [ ] [CoL]
 { int row, col;
    for ( row = 0; row < Row; row++)
     {
     for (col = 0; col < CoL ; col++) {
     printf ( "%d" , *(*(mat + row) + col)); }
     printf ("\n" );
     }
 }
void matrix multiply (int mat1 [ ] [CoL],
                      int mat2 [ ] [CoL],
                      int res [ ] [CoL])
{ int row, col , i, sum;
  for (row = 0; row < Row; row++)
   {
    for ( col = 0; col < CoL; col++)
     { sum = 0;
       for (i = 0; i < CoL; i++)
        {
        sum += (*(*(mat1 + row) + i)) * (*(*(mat2 + i)
                                          col));
}
```

```
        *(*(res + row) + col) = sum;
      }
    }
}
int main()
{ int mat1[Row][Col], mat2[Row][Col],
      product[Row][Col];
  printf("Enter first matrix of size 3X3: ");
  matrixinput(mat1);
  printf("enter second matrix of size 3X3: ");
  matrixinput(mat2);
  matrixmultiply(mat1, mat2, product);
  printf("product of both matrix: \n");
  matrixprint(product);
  return 0;
}
```

Output →

```
enter first matrix of size 3X3:
1 0 0 0 1 0 0 0 1
enter second matrix of size 3X3:
1 0 0 0 1 0 0 0 1
product of both matrix:
1 0 0
0 1 0
0 0 1
```

7

a) 2, 5, 6, 8, 10,

b) 20, 9, 9

c) 0

d) 25