

Practical Workbook

CT-175

**PROGRAMMING
FUNDAMENTALS**



Name: Hamna Ali Khan

Year: 2024

Batch: 2028

Roll No: CT-157

Department: BCIT

Dept. of Computer Science & Information Technology
NED University of Engineering & Technology

EXERCISE Q# 01

As a programmer, you are required to create a program that takes the first and last name from a user. The program then combines both the inputs taken and prints the string backwards.

```
#include<stdio.h>
#include<string.h>
int main(){
char firstname[100];
char lastname[100];
char fullname[200];
printf("Enter your first name: ");
scanf("%s", firstname);
printf("Enter your last name: ");
scanf("%s", lastname);

strcpy(fullname,firstname);
strcat(fullname," ");
strcat(fullname,lastname);
printf("Full name is = %s\n", fullname);

//printing backward string
printf("\n-----\n");
int length= strlen(fullname);
printf("Your name in reverse: ");

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

    printf("%c", fullname[i]);

}
printf("\n");

return 0;

}
```

OUTPUT:

```
Enter your first name: Hamna
Enter your last name: Ali
Full name is = Hamna Ali

-----
Your name in reverse: ilA anmaH
```

EXERCISE Q# 02

Each student is required to find out the maximum frequency of characters occurring in their name and the courses offered in Fall 2021. To find it, the student enters their name, courses offered and the program finds the maximum occurrences of a character in the name and course. Course names should be used like Programming Fundamentals, Applied Physics, Pakistan Studies and so on.

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#define max_len 100

int findmaxfreq(char str[]){
    int freq[256] = {0};
    int maxfreq = 0;

    for(int i=0;str[i]!='\0';i++){
        if(!isspace(str[i])){
            freq[(unsigned char)str[i]]++;
            if(freq[(unsigned char)str[i]]>maxfreq){
                maxfreq=freq[(unsigned char)str[i]];
            }
        }
    }
    return maxfreq;
}

int main(){
    char name[max_len];
    char courses[max_len];

    printf("Enter your name: ");
    fgets(name,max_len,stdin);
    name[strcspn(name, "\n")] = '\0';

    printf("Enter no of courses(separated by commas): ");
    fgets(courses,max_len,stdin);
    courses[strcspn(courses, "\n")] = '\0';
```

```
int maxfreqname= findmaxfreq(name);
int maxfreqcourses= findmaxfreq(courses);

printf("\nMaximum frequency of characters: ");
printf("In name: %d\n", maxfreqname);
printf("In courses: %d\n", maxfreqcourses);
return 0;
}
```

OUTPUT:

```
Enter your name: hamna ali khan
Enter the names of the courses offered (separated by commas): Programming Fundamentals, Applied Physics, Pakistan Studies

Maximum frequency of characters:
In name: 4
In courses: 5
```

EXERCISE Q# 03

Students are grouped in two to complete a lab task. Each student is required to enter a string of their own choice as an input to the program. The program will then display as a result whether both the strings are equal. If the strings are not equal, the program will display which of the string is greater.

Test cases:

Enter two strings that are same.

Enter two different strings.

```
#include<stdio.h>
#include<string.h>
int main(){
char str1[100];
char str2[100];

printf("Enter first string: ");
fgets(str1,sizeof(str1),stdin);
str1[strcspn(str1,"\n")]='\0';
printf("Enter second string: ");
fgets(str2,sizeof(str2),stdin);
str2[strcspn(str2,"\n")]='\0';

int result;
result=strcmp(str1,str2);
if(result==0){
printf("\nThe strings are equal");
}else if(result>0){
```

```
printf("%s' string is greater than '%s' string",str1,str2);

}else{
    printf("%s' string is less than '%s' string",str1,str2);

}

return 0;
}
```

OUTPUT:

```
Enter first string: Hi! I'm Hamna
Enter second string: Nice to meet you!
'Hi! I'm Hamna' string is less than 'Nice to meet you!' string
```

OUTPUT:

```
Enter first string: Hi! I'm Hamna
Enter second string: Hi! I'm Hamna

The strings are equal
```

EXERCISE Q# 04

4. Write down the output of the following program.

```
int main (void)
{
    char a[11] = "hello world";
    int i;
    for(i = 0; i <= 9; i++)
    {
        a[i] = a[i + 1];
        printf("%d \t %s \n", i, a);
    }
    Printf("\n %d", a);}
```

ERRORS:

- First the string a has 11 characters including the null character but when initializes it does not have room for that null character.

-Second we have, The loop runs from i=0 to i<=9, modifying a[i]=a[i+1] in each iteration. This overwrites the last character and shifts the null terminator outside of the valid array bounds, causing issues.

-Thirdly, the char array is defined by %d instead of %s plus Printf is written which is wrong. We have to write printf.

OUTPUT:

```
0      ello world
1      llo world
2      lo world
3      o world
4      world
5      world
6      orld
7      rld
8      ld
9      d
```

After the loop, the null terminator has been lost, and the array a no longer forms a proper string. The %d format specifier treats a as a pointer to its memory address, so it will print an integer representation of the address (e.g., 6422276). This is not meaningful and highlights the problem with the code.

Conclusion:

The program as written is incorrect, and the output is unpredictable due to undefined behavior. A corrected version is essential for reliable results

