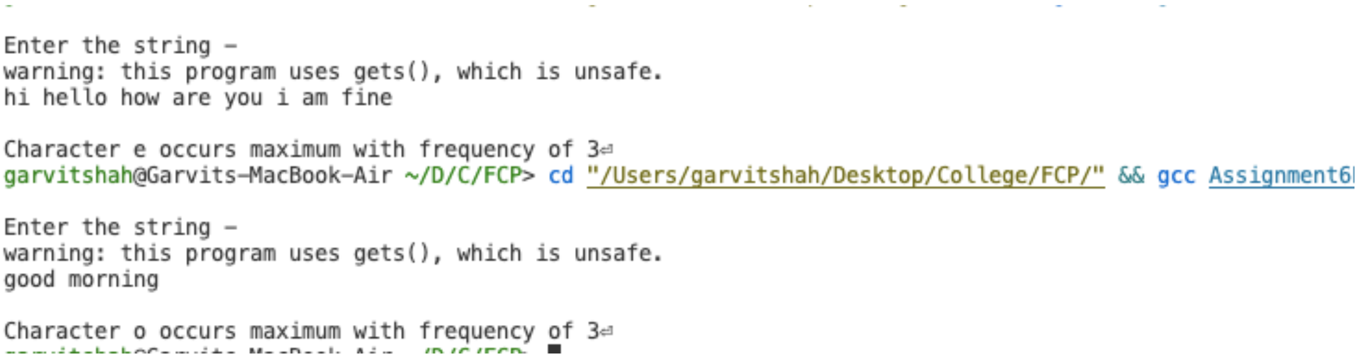
Assignment 6: String [Extended]

Garvit Shah
F24
U21CS089

1. Write a program in C to find the maximum occurring character in a string.

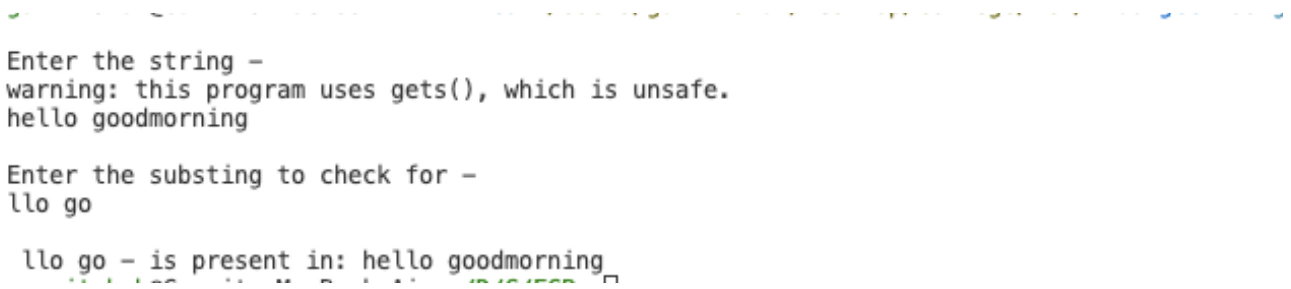
```
//Q1
#include <stdio.h>
#include<ctype.h>
int main(){
    int i=0, n=0, max1=0, max2=0, max=0, j=0, p, q;
    int hash[256];
    char str[100], a;
    printf("\nEnter the string - \n");
    gets(str);
    for(i=0;i<256;i++){
        hash[i] = 0;
    }
    i=0;
    while(str[i] != '\0'){
        n = str[i];
        if (hash[n] < 1){
            hash[n] = 1;
        }
        else{
           hash[n] += 1;
        }
        i+=1;
    }

    for(i=65;i<=90; i++){
        if(hash[i] > max1){
            max1 = hash[i];
            p=i;
        }
    }

    for(i=97;i<=122; i++){
        if(hash[i] > max2){
            max2 = hash[i];
            q=i;
        }
    }

    if(max1 > max2){
        max = max1;
        a = p;
    }
    else{
        max = max2;
        a = q;
    }
    printf("\nCharacter %c occurs maximum with frequency of %d", a,
max);
}
```



```
Enter the string -
warning: this program uses gets(), which is unsafe.
hi hello how are you i am fine

Character e occurs maximum with frequency of 3⏎
garvitshah@Garvits-MacBook-Air ~/D/C/FCP> cd "/Users/garvitshah/Desktop/College/FCP/" && gcc Assignment6

Enter the string -
warning: this program uses gets(), which is unsafe.
good morning

Character o occurs maximum with frequency of 3⏎
garvitshah@Garvits-MacBook-Air ~/D/C/FCP> █
```

2. Write a C program to check whether a given substring is present in the given string.

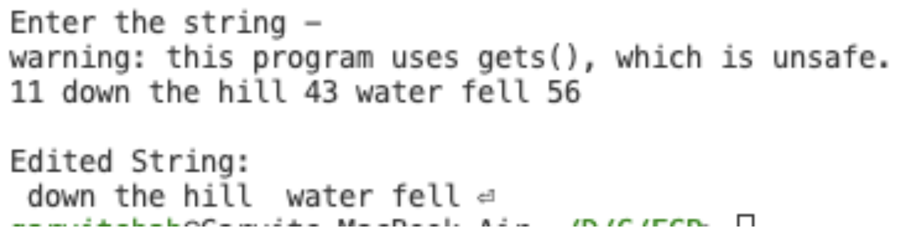
```c
//Q2
#include <stdio.h>
int main(){
    char str1[200], str2[100];
    int i=0, j=0, k=0, l2=0;
    printf("\nEnter the string - \n");
    gets(str1);
    printf("\nEnter the substing to check for - \n");
    gets(str2);
    while(str2[l2] != '\0'){
        l2+=1;
    }
    while(str1[i] != '\0'){
        j=0;
        k=0;
        while(str1[i] == str2[j]){
            k+=1;
            i+=1;
            j+=1;
        }
        if (k == l2){
            printf("\n %s - is present in: %s \n", str2, str1);
        }
        i+=1;
    }
}
```



```
Enter the string -
warning: this program uses gets(), which is unsafe.
hello goodmorning

Enter the substing to check for -
llo go

 llo go - is present in: hello goodmorning
```

3. **Write a program in C to remove only digits from the string if any(exclude alphabets).**

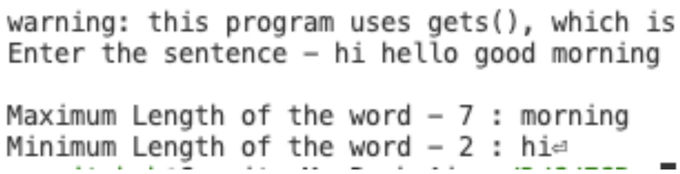
```c
//Q3
#include <stdio.h>
#include <ctype.h>
int main(){
    char str1[100], str2[100];
    int i=0, j=0, k=0;
    printf("\nEnter the string - \n");
    gets(str1);
    while(str1[i] != '\0'){
        if (isdigit(str1[i]) == 0){
            k = 0;
            str2[j] = str1[i];
            j+=1;
            i+=1;
        }
        else{
            i+=1;
            k+=1;
        }
    }
    printf("\nEdited String: \n%s", str2);
}
```



```
Enter the string -
warning: this program uses gets(), which is unsafe.
11 down the hill 43 water fell 56

Edited String:
 down the hill  water fell ⏎
```

4. Write a program in C to find the largest and smallest word in a string.

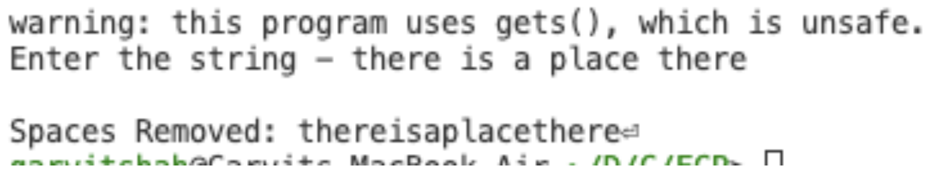
```c
//Q4
#include <stdio.h>
int main(){
    int i=0, j=0, max=0, min=100, k=0, n=0;
    char str1[100], Mword[50], mword[50];
    printf("\nEnter the sentence - ");
    gets(str1);
    while(str1[i] != '\0'){
        k=0;

        while((str1[i] != ' ') &&(str1[i] != '\0')){
            k+=1;
            i+=1;
        }

        // printf("\n%d : %c", k, str1[i-1]);
        if (k > max){
            max = k;
            for(n=0;n<max;n++){
                Mword[n] = str1[i-k];
                printf("\n%c, %d, %c", Mword[n], i-k, str1[i-k]);
                i+=1;
            }
            Mword[n] = '\0';
            i-=k;
        }
        printf("\n%s", Mword);

        if ((k< min)&&(k!=0)){
            min = k;
            for(n=0;n<min;n++){
                mword[n] = str1[i-k];
                i+=1;
            }
            mword[n] = '\0';
            i-=k;
        }
        i+=1;
    }
    printf("\nMaximum Length of the word - %d : %s\nMinimum Length of
the word - %d : %s", max, Mword, min, mword);
}
```



```
warning: this program uses gets(), which is unsafe.
Enter the sentence - hi hello good morning

Maximum Length of the word - 7 : morning
Minimum Length of the word - 2 : hi⏎
```

5. **Write a program in C to read a file and remove the spaces between two words of its content.**

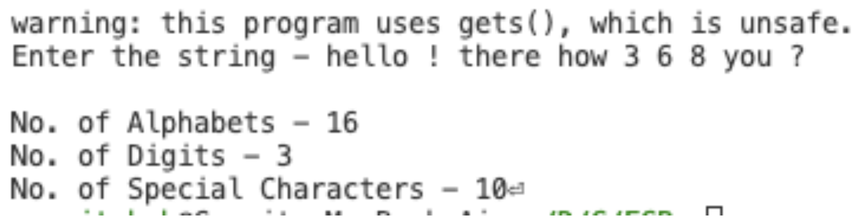
```
//Q5
#include <stdio.h>
int main(){
    int i=0, j=0;
    char str1[100], str2[100];
    printf("\nEnter the string - ");
    gets(str1);
    while(str1[i] != '\0'){
        if((str1[i] != ' ') && (str1[i] != '\0')){
            str2[j] = str1[i];
            i+=1;
            j+=1;
        }
        else{
            i+=1;
        }
    }
    printf("\nSpaces Removed: %s", str2);
}
```



```
warning: this program uses gets(), which is unsafe.
Enter the string - there is a place there

Spaces Removed: thereisaplacethere⏎
garvitshah@Garvits-MacBook-Air ~/D/C/FCP> □
```

6. **Write a program in C to count the total number of alphabets, digits and special characters in a string.**

```c
//Q6
#include <stdio.h>
#include <ctype.h>
int main(){
    char str1[100];
    int arr1[3], i=0;
    printf("\nEnter the string - ");
    gets(str1);
    arr1[0] = 0;
    arr1[1] = 0;
    arr1[2] = 0;
    while(str1[i]!='\0'){
        if (isdigit(str1[i]) == 1){
            arr1[1] +=1;
        }
        else if(((str1[i]>=65) && (str1[i] <=90)) || ((str1[i]>=97) &&
(str1[i] <=122))){
            arr1[0] +=1;
        }
        else{
            arr1[2] +=1;
        }
        i+=1;
    }
    printf("\nNo. of Alphabets - %d\nNo. of Digits - %d\nNo. of Special
Characters - %d", arr1[0], arr1[1], arr1[2]);
}
```



```
warning: this program uses gets(), which is unsafe.
Enter the string - hello ! there how 3 6 8 you ?

No. of Alphabets - 16
No. of Digits - 3
No. of Special Characters - 10⏎
```