Char array

1

```
#include <iostream>
#include <cstring>
using namespace std;
int main(){
    char a[] = {'a','b','c','t','r','q','e','y','\0'};
    //or
    char a2[] = "ashish";
    cout << a2 << endl;
    cout << a << endl;
    cout << strlen(a) <<endl; //no of visible char
    cout << sizeof(a) << endl; // +1 because of the null char

}
```

2

```
#include <iostream>
using namespace std;
int main(){
    char sentence[1000];
    char temp = cin.get();
    int len = 0;
    while(temp!='#'){
        sentence[len++] = temp;
        temp = cin.get();
    }
    sentence[len] = '\0';
    cout << "Length" << len << endl;
    cout << sentence << endl;
}
```

3\

```
#include <iostream>
using namespace std;
//given a sentence count the number of alphabets , digits ,
//spaces in the sentence
int main(){
    //store and then count
    //read one by one and then count
    char ch;
    ch = cin.get();
    //count
    int alpha = 0;
    int space = 0;
```

```
    int digit = 0;
    while(ch!='\n'){
        if(ch>='o' and ch<='9'){
            digit++;
        }
        else if((ch>='a' and ch<='z') or (ch>='A' and ch<='Z')){
            alpha++;
        }
        else if(ch==' ' or ch=='\t'){
            space++;
        }
        ch = cin.get();
    }
    cout << "total alphabets " << alpha << endl;
    cout << "total digit" << digit << endl;
    cout << "total space" << space << endl;
    return 0;
}
```

4

```
#include <iostream>
using namespace std;
int main(){
    char sentence[1000];
    cin.getline(sentence,1000, '#'); //if we have to set like stop at # then
add third para #
    cout << sentence << endl;
}
```

5

```
//given a route containing N S E W directions find the sortest path to reach
the location
#include <iostream>
using namespace std;
int main(){
    char route[1000];
    cin.getline(route,1000);
    int x = 0;
    int y = 0;
    for (int i = 0; route[i] != '\0'; i++)
    {
        switch(route[i]){
            case 'N' :y++;
                      break;
            case 'S' :y--;
                      break;
```

```
            case 'E' :x++;
                     break;
            case 'w' :x--;
                     break;
        }
    }
    cout << "final X :" <<x <<" and y :" <<y <<endl;
    //other case
    if(x>=0 and y>=0){
        while(y--){
            cout<<"N";
        }
        while(x--){
            cout<<"E";
        }
    cout<<"  NEW CONDITION"<<endl;
    }
    if(x<0 and y>0){
        while(y--){
            cout<<"N";
        }
        while(x++){
            cout<<"W";
        }
        cout<<"    NEW CONDITION"<<endl;
    }
    if (x<0 and y<0)
    {
        while(y++){
            cout<<"S";
        }
        while(x++){
            cout<<"W";
        }
        cout<<"    NEW CONDITION"<<endl;
    }
    if(x>0 and y<0){
        while(y++){
            cout<<"S";
        }
        while(x--){
            cout<<"E";
        }
        cout<<"   NEW CONDITION"<<endl;
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    char a[1000] = "1ewgle";
    char b[1000];
    //calc length
    cout << strlen(a) << endl;
    //str copy
    strcpy(b,a);
    cout << b << endl;
    //strcat
    char web[] ="www.";
    char domain[] = "google.com";
    cout << strcat(web,domain) << endl;
    //str compare
    cout << strcmp(a,b) << endl;
    cout <<strcmp(web,domain) << endl;

}
```

\

7

```cpp
//problem : Read N strings and print the largest string.
//each string can have upto 1000 characters
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin >> n;
    cin.get();
    char sentence[1000];
    char largest[1000];
    int largest_len = 0;
    while(n--){
        cin.getline(sentence,1000);
        cout << sentence << endl;
        int len = strlen(sentence);
        if(len > largest_len){
            largest_len = len;
            strcpy(largest,sentence);
        }
    }
    cout << "Largest sentence is :" << largest << endl;
    return 0;
}
```

8

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    //char s[1000] = {'1','2','4','3','a' ,'e','\0'};
    string s; //= "hello world"; //dynamic array
    getline(cin,s,'.');
    for(char ch : s){
        cout << ch << ",";

}   cout << s << endl; //or


}
```

9

```
//run length encoding for string ompression
//if compressed string is bigger than orignal return original string
#include <iostream>
using namespace std;
string compressString(string str){
    int n = str.length();
    string output;
    for (int i = 0; i < n; ++i)
    {
        int count = 1;
        while(i <n-1 and str[i+1] == str[i]){
            count++;
            i++;
        }
        output += str[i];
        output += to_string(count);
        //complexity is linear because we incrementing the i in while loop
also
    }
    if(output.length() > n){
        return str;
    }
    return output;
}
int main(){
    string s1 = "aaabbccddee";
    cout << compressString(s1) <<endl;
    string s2 = "abcdefg";
    cout << compressString(s2) << endl;
    return 0;
```

```
}
```