Strings

→ Strings are basically character arrays

Declaration of strings

```
string str = "Aniruddh";
cout << str;
```

→ str[0] = A.

⇒ str = {'A', 'n', 'i', 'r', 'u', 'd', 'h', '\0'}

'\0' → Null Character

Taking input

(*)
```
string s;
cin >> s;
```
→ It will not accept the string after first space.

→ To input a string with spaces, we have to use "getline(cin, s);"

Eg-
```
string name;
getline(cin, name);
```

Count vowels in a string

→ Next page

Date : / /
Page No.

```
int main (){
    string s = "Aniruudh";
    int count = 0
    int i = 0;
    while (str[i] != '\0') {
        if (str[i]=='a' || str[i]=='e' || str[i]=='i'||
            str[i]=='o' || str[i]=='u') {
            count++;
        } i++;
    }
    cout<<count ;
}
```

⇒ Updation of single character in string -

```
0 1 2 3 4 5 6
string str = "Anirudh"
cout<<str; // Aniruudh
str[2] = 'o';
cout<<str; // Anoruudh
```

Note → In C++ strings are mutable.

⇒ Built in string functions

① <string_name>.size();
↳ This will not include the null character '\0'
→ It will give us the length of string.

② push_back();

```
string str = "abcd";
str.push_back('e'); // str becomes "abcde"
```

Date : / /
Page No.

→ push_back( ) can only insert one character.

③ pop_back( );

↳ Pops out the last character.

⇒ "+" operator

```
string s="abc";
s=s+"xyz"; //append or modify
cout<<s<<endl; //abcxyz
```

→ reverse ( )

```
string str="abcdef";
(*) reverse (str.begin(), str.end());
cout<<str; // fedcba
(*) reverse (s.begin()+2, s.end());
cout<<str; // abfedc
(*) reverse (str.begin()+2, str.begin()+5);   index+1
cout<<str; // abedcf
```

Q) Input a string of even length and reverse the first half of the string

I/P → abcdefgh
O/P → dcbaefgh

Sol") reverse n = (string_name).size()
reverse (s.begin(), s.begin()+n/2);

Date : / /
Page No.

→ substr()

```
int main(){
    string s = "abcdef";
    cout << s.substr(0); // abcdef
    cout << s.substr(1); // bcdef
    cout << s.substr(1,3); bcd
```

→ to_string()

→ int main(){
    int x = 1234;
    string s = to_string(x);
                ↳ convert x to string
}

⇒ Sorting a string using in-built function

```
string s = "name";
sort(s.begin(), s.end());
//s → aemn
```

It will sort the string in order of their ascii value including special characters.

Q Given a string consisting of lowercase English alphabets. Print the character that is occuring most number of times.

Date : / /
Page No.

→ string S = "hello";

h → 1
e → 1
l → 2
o → 1

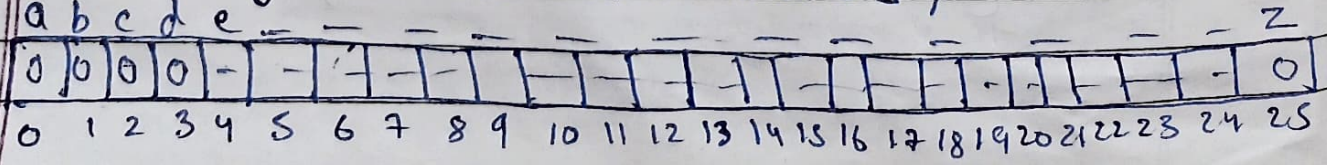
→ O/P → 2

⇒ brute force →

```
int main(){
    string S = "hello";
    maxCount = 0;
    for(int i=0; i<S.size(); i++){
        char ch = S[i];
        int count = 1;
        for(int j=i+1; j<S.size(); j++){
            if(S[j] == ch) count++;
        }
        if(maxCount < count) max = count;
    }
    cout << maxCount;
}
```

⇒ Efficient method

→ Here we have to use a special array → int arr[26];



Date : / /
Page No.

→ mapping the indexes with alphabets

```
int arr[26];
for(int i=0; i<s.length(); i++){
    char ch=s[i];
    int ascii=(int)ch;
    arr[ascii-97]++;
}
int max=0
for(int i=0; i<26; i++){
    if(arr[i]>max) max=arr[i];
}
for(int i=0; i<26; i++){
    if(arr[i]==max){
        int ascii=i+97;
        char ch=(char)ascii;
        cout<<ch<<" "<<max<<endl;
    }
}
```

⇒ Stringstream class

→ #include<sstream> → This header file has to be included to use stringstream class

```
int main(){
    string str="My name is Anvindh";
    stringstream ss(str);
    string temp;
```

Date : / /
Page No.

```
while (ss>> temp){
    cout << temp << endl;
}
```

→ O/P → My
name
is
Anirudh

8) Given a sentence 'str', return the word that is occuring most number of times in that sentence.

→
```
int main (){
    string str = "How is it going. My name is ABC";
    stringstream ss(str);
    string temp;
    vector <string> v;
    while (ss>> temp){
        v.push_back(temp);
    }

    sort (v.begin(), v.end());
    int maxCount=1, count=1;
    for (int i=1; i<v.size(); i++){
        if (v[i] = v[i-1]) count++;
        else count=1;
        maxCount = max(maxCount, count);
    } count=1;
    for (int i=1; i<v.size(); i++){
        if (v[i] == v[i-1]) count++;
        else count=1;
```

```
        if (count == maxCount) {
            cout << v[i] << "  " << endl;
        }
    }
}
```

→ stoi VS stoll built in functions

↓ string to integer → -2^{31} to $2^{31}-1$

string to long long → -2^{63} to $2^{63}-1$

```
→ int main() {
    string str = "123456";
    int x = stoi(str);
    cout << x+1; //123457
}
```

→ ∞ To store big numbers we can use stoll(str);.

→ Vectors

Q) Given n strings consisting of digits from 0 to 9. Return the index of string which has maximum value. (Take 0 based indexing)

0 1 2 3 4 5

→ I/P: 0123, 0023, 456, 00182, 940, 2901

O/P: 5

→ Convert the strings to integers and run a loop to find the max element.