|  |  |  |
| --- | --- | --- |
| Write an application in C++ using to use new and delete to manage dynamic memory. Implement the application by defining a suitable class and its member functions. | | |
| Requirement Tag | Requirement Description | Comments |
| OSTR/01 | Accept a sentence (that is a line) with a maximum of 5 words from the user. |  |
| OSTR/02 | Extract every word and store in an array of 5 char pointers dynamically allocating memory. | Hint: Use new (nothrow) and handle new allocation failures manually. Refer http://www.cplusplus.com/doc/tutorial/dynamic/ |
| OSTR/03 | Print the longest word in the sentence (assume all words are of different length). |  |
| OSTR/04 | Read a word and a replace word from user. Perform search-replace. (Replace only the first matching word). | Hint: While replacing, free the memory allocated for the searched word and then allocate new memory as per length of replace word. |
| OSTR/05 | Display all the words. |  |
| OSTR/06 | Free all the allocated memory on exit |  |

1. Accept a sentence (that is a line) with a maximum of 5 words from the user

#include <iostream>

using namespace std;

int main(){

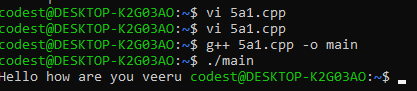
char arr[][50]={"Hello","how","are","you","veeru"};

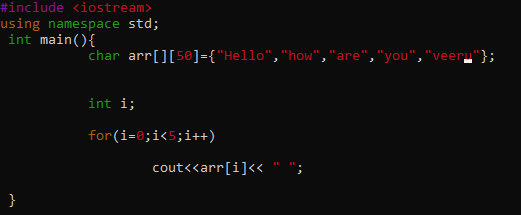
int i;

for(i=0;i<5;i++)

cout<<arr[i]<< " ";

}





2. Extract every word and store in an array of 5 char pointers dynamically allocating memory.

#include <iostream>

using namespace std;

int main(){

string \*s = new string[50];

int i;

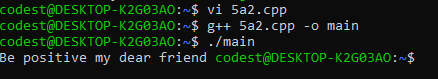
for(i=0;i<5;i++){

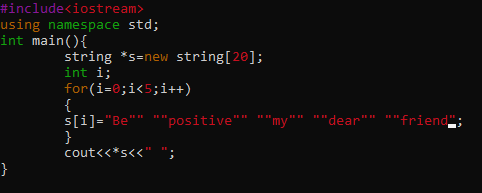
s [i]="Hello" " " "how" " " "are"" " "you"" " "veeru";

}

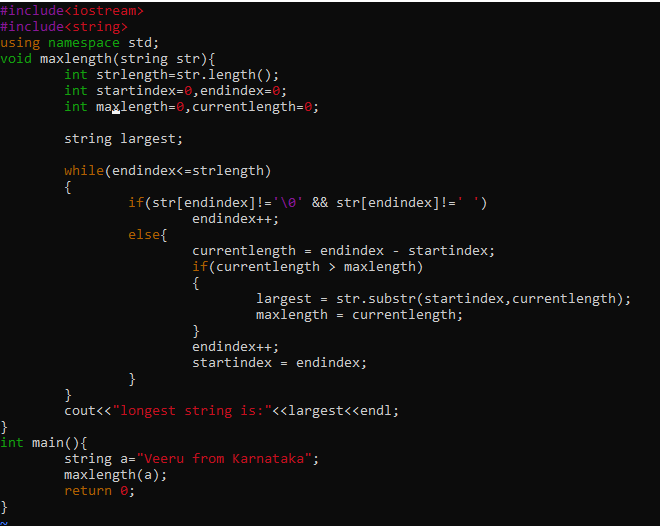
cout<<\*s<< " ";

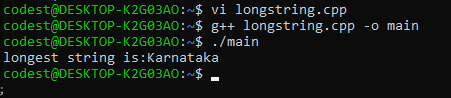
}



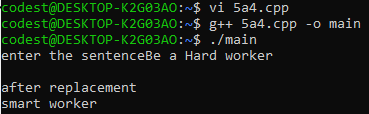


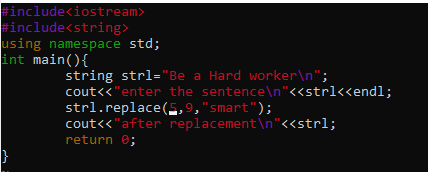
3. Print the longest word in the sentence (assume all words are of different lengths).





4. Read a word and replace the word from the user. Perform search-replace. (Replace only the first matching word).





6.Free all the allocated memory on exit

Ans: use syntax free (ptr);