TASK-6

#include<bits/stdc++.h>

using namespace std;

class MyString

{

char\* c;

public:

MyString()

{

c = NULL;

}

int getlen()

{

int len;

for(int i=0;;i++)

{

if(c[i]==NULL)

{

len = i;

break;

}

}

return len;

}

MyString(char\* in\_char)

{

int idx = 0;

while (in\_char[idx] != '\0')

{

idx++;

}

c = (char\*) malloc(idx\*sizeof(char));

for(int i=0; i<idx; i++)

{

c[i]=in\_char[i];

}

c[idx]='\0';

}

void append(MyString s)

{

int newLen = s.getlen() + getlen();

char\* str = (char\*) malloc(newLen\*sizeof(char));

int idx = 0;

for (int i= 0; i< getlen(); i++)

{

str[idx++] = c[i];

}

for (int i= 0; i< s.getlen(); i++)

{

str[idx++] = s.c[i];

}

str[idx] = '\0';

c = str;

}

void append(char\* in\_char)

{

MyString s(in\_char);

int newLen = s.getlen() + getlen();

char\* str = (char\*) malloc(newLen\*sizeof(char));

int idx = 0;

for (int i= 0; i< getlen(); i++)

{

str[idx++] = c[i];

}

for (int i= 0; i< s.getlen(); i++)

{

str[idx++] = s.c[i];

}

str[idx] = '\0';

c = str;

}

MyString operator+(MyString s)

{

MyString temp(c);

temp.append(s);

return temp;

}

friend MyString operator+(char\* cha, MyString s)

{

MyString temp(cha);

temp.append(s);

return temp;

}

//equal is overloaded by default copy constructor

void operator+=(MyString s)

{

append(s);

}

bool operator!()

{

if(getlen()==0)

return true;

else

return false;

}

char operator[](int i)

{

return c[i];

}

void display()

{

for(int i=0; i<getlen(); i++)

{

cout<<c[i];

}

}

};