# Question 6: String

**Q6.1: Print a string “Hello world” to screen, by using:**

* **char \*st = “Hello world”**
* **char st[] = {‘H’, ‘e’, ….}**
* **char \*st = new char[…] ….**

**Code:**

#include<iostream>

using namespace std;

int main() {

char \*st1 = "Hello world";

char st2[] = { 'H','e','l','l','o',' ','w','o','r','l','d','\0'};

char \*st3 = new char[12];

//c1

//memcpy(st3, "Hello world", 12);

//c2

strcpy\_s(st3,sizeof("Hello world"), "Hello world");

cout << st1 << endl;

cout << st2 << endl;

cout << st3 << endl;

return 0;

}

**Q6.2: Implement a function to clone a string:**

void clone(char \*input, char \*output)

#include<iostream>

using namespace std;

void clone(char \*input, char \*output);

int main() {

char \*a = "chuong np", b[256];

clone(a, b);

cout << a << endl << b << endl;

return 0;

}

void clone(char \* input, char \* output)

{

int n = strlen(input);

for (int i = 0; i <= n; i++) {

output[i] = input[i];

}

}

**Q6.3: Implement a function to copy a string:**

**void copyString(char \*input, int offset, int length, bool invert, char \*output, int output\_offset)**

* + **input: input string**
  + **offset: starting position for copy**
  + **length: length of substring to be copied**
  + **invert: invert the result**
  + **output: output string**
  + **output\_offset: the beginning of copied position of output**

**for example:**

**st = “Hello world”**

**st2 = “My name is C++”**

**copyString(st, 6, 5, true, st2, 11) 🡪 st2 = “My name is dlr”;**

**Implement this function in 3 ways:**

* **using memcpy**
* **using strcpy**
* **without memcpy and strcpy.**

**Code:**

#include<iostream>

#include<string>

using namespace std;

#pragma warning(disable : 4996)

// bo qua yeu cau dung ham strncpy\_s()

#define Min(m,n) m>n?n:m

void copyString\_memcpy(char \*input, int offset, int lenght, bool invert, char \*output, int output\_offset);

void copyString\_strcpy(char \*input, int offset, int lenght, bool invert, char \*output, int output\_offset);

void copyString(char \*input, int offset, int lenght, bool invert, char \*output, int output\_offset);

void Swap(char &a, char &b);

int main() {

char st[] = "Hello world", st2[] = "My name is C++";

cout << st << endl;

cout << st2 << endl;

copyString\_strcpy(st, 6, 5, false, st2, 11);

cout << st2 << endl;

return 0;

}

void copyString\_memcpy(char \* input, int offset, int lenght, bool invert, char \* output, int output\_offset)

{

if (invert) {

//copy phan cuoi cua chuoi

int n = (strlen(output) - output\_offset);

n = (Min(n,lenght));

int m = (strlen(input) - offset);

memcpy(output + output\_offset, input + offset + m - (Min(m, n)), Min(m, n));

// dao chuoi

for (register int i = 0; i <= (Min(m, n))/2; i++) {

Swap(output[output\_offset + i], output[output\_offset + n - i-1]);

}

}

else {

// số byte cần copy phải nhỏ hơn số byte tính từ output\_offset đến kí tự '/0'

int n = (strlen(output) - output\_offset);

n = n > lenght ? lenght : n;

int m = (strlen(input) - offset);

memcpy(output + output\_offset, input + offset, Min(m, n));

}

}

void copyString\_strcpy(char \* input, int offset, int lenght, bool invert, char \*output, int output\_offset)

{

if (invert) {

//copy phan cuoi cua chuoi

int n = (strlen(output) - output\_offset);

n = (Min(n, lenght));

int m = (strlen(input) - offset);

strncpy(output + output\_offset, input + offset + m - (Min(m, n)), Min(m, n));

// dao chuoi

for (register int i = 0; i <= (Min(m, n)) / 2; i++) {

Swap(output[output\_offset + i], output[output\_offset + n - i - 1]);

}

}

else {

int n = strlen(output) - output\_offset;

n = (Min(n, lenght));

int m = strlen(input) - offset;

cout << strlen(output) << endl;

strncpy(output + output\_offset, input + offset, Min(m, n));

}

}

void copyString(char \* input, int offset, int lenght, bool invert, char \* output, int output\_offset)

{

if (invert) {

for (register int i = 0; i < lenght; i++) {

if (output[output\_offset + lenght - 1 - i] && input[offset + i])

output[output\_offset + lenght - 1 - i] = input[offset + i];

else

break;

}

}

else {

for (register int i = 0; i < lenght; i++) {

if (output[output\_offset + i] && input[offset + i])

output[output\_offset + i] = input[offset + i];

else

break;

}

}

}

void Swap(char &a, char &b)

{

char temp = a;

a = b;

b = temp;

}