Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Ba, 11 tháng 2 2025, 1:22 PM
Kết thúc lúc	Chủ Nhật, 23 tháng 2 2025, 11:46 PM
Thời gian thực hiện	12 Các ngày 10 giờ
Điểm	12,00/12,00
Điểm	**10,00** trên 10,00 (**100**%)

Câu hỏi 1

Đúng

Đạt điểm 1,00 trên 1,00

Implement function

```
void printArray(int n){}
```

to print 0, 1, 2, ..., n (n is positive integer and has no space at the end).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`printArray(5);`	`0, 1, 2, 3, 4, 5`
`printArray(10);`	`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10`

Answer: (penalty regime: 0, 0, 0, 5, 10, 15, ... %)

Reset answer

```
void printArrayHelper(int current, int n) {
    if (current > n) return;
    cout << current;
    if (current < n) cout << ", ";
    printArrayHelper(current + 1, n);
}

void printArray(int n) {
    printArrayHelper(0, n);
}
```

	Test	Expected	Got	
✓	`printArray(5);`	`0, 1, 2, 3, 4, 5`	`0, 1, 2, 3, 4, 5`	✓
✓	`printArray(10);`	`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10`	`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10`	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **2**

Đúng

Đạt điểm 1,00 trên 1,00

Given a positive number, print following a pattern without using any loop.

```
Input: n = 16
Output: 16, 11, 6, 1, -4, 1, 6, 11, 16 (has no space at the end)

Input: n = 10
Output: 10, 5, 0, 5, 10 (has no space at the end)
```

We basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or negative, we one add 5 until we reach n.

Note: Please note that you can't using key work for, while, goto (even in variable names, comment).

You can implement other recursive functions if needed.

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`printPattern(14);`	`14 9 4 -1 4 9 14`

Answer: (penalty regime: 0 %)

Reset answer

```
void printPattern(int n) {
    cout << n;
    if (n > 0) {
        cout << " ";
        printPattern(n - 5);
        cout << " " << n;
    }
}
```

	Test	Expected	Got	
✓	`printPattern(14);`	`14 9 4 -1 4 9 14`	`14 9 4 -1 4 9 14`	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **3**

Đúng

Đạt điểm 1,00 trên 1,00

Implement function

```
int findMax(int* arr, int length){}
```

to find the largest element using recursion (with length is the number of elements in integer array arr).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result

int arr[] = {10, 5, 7, 9, 15, 6, 11, 8, 12, 2};
cout << findMax(arr, 10);
	15

Answer: (penalty regime: 0, 0, 0, 5, 10, ... %)

Reset answer

```
int findMax(int* arr, int length) {
    if (length == 1) return arr[0];
    int maxRest = findMax(arr + 1, length - 1);
    return (arr[0] > maxRest) ? arr[0] : maxRest;
}
```

	Test	Expected	Got	

✓	int arr[] = {10, 5, 7, 9, 15, 6, 11, 8, 12, 2};
	cout << findMax(arr, 10);
		15	15	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **4**

Đúng

Đạt điểm 1,00 trên 1,00

Implement function

```
bool isPalindrome(string str){}
```

to check if the given non empty string is palindrome, else not palindrome using recursion.

In test case, for extra point, we will have some palindrome sentences (All remaining test cases are words).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream>, #include <string.h> and using namespace std;

For example:

Test	Result
`cout << isPalindrome("mom");`	1
`cout << isPalindrome("do geese see god");`	1

Answer: (penalty regime: 0 %)

Reset answer

```
bool isAlpha(char c) {
    return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
}

char toLower(char c) {
    return (c >= 'A' && c <= 'Z') ? c + ('a' - 'A') : c;
}

bool checkPalindrome(const string &str, int left, int right) {
    if (left >= right) return true;

    if (!isAlpha(str[left])) return checkPalindrome(str, left + 1, right);
    if (!isAlpha(str[right])) return checkPalindrome(str, left, right - 1);

    if (toLower(str[left]) != toLower(str[right])) return false;

    return checkPalindrome(str, left + 1, right - 1);
}

bool isPalindrome(string str) {
    return checkPalindrome(str, 0, static_cast<int>(str.length()) - 1);
}

```

	Test	Expected	Got	
✓	`cout << isPalindrome("mom");`	1	1	✓

	Test	Expected	Got	
✓	`cout << isPalindrome("do geese see god");`	1	1	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **5**

Đúng

Đạt điểm 1,00 trên 1,00

Give two positive integers a and b, implement function

```
int findGCD(int a, int b){}
```

to find **GCD** (Greatest Common Divisor) of a and b using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`cout << findGCD(124,32);`	4

Answer: (penalty regime: 0 %)

Reset answer

```
int findGCD(int a, int b) {
    if (b == 0) return a;
    return findGCD(b, a % b);
}
```

	Test	Expected	Got	
✓	`cout << findGCD(124,32);`	4	4	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **6**

Đúng

Đạt điểm 1,00 trên 1,00

String s contains lowercase letters, digits, "(" and ")", satisfying the following rules:

- Two digits cannot be adjacent.
- Two "(" cannot be adjacent.
- One "(" and one ")" cannot be adjacent.
- After any digit, there must be "(".
- The quantities of "(" and ")" are equal.

Change string s until new string t created, t contains only lowercase letters. These are changing rules:

- Sub-strings with form "n(p)", can change to "pp...p" (n times p), where n is a digit and p is a string.
- If p still contains "(", ")" or digits, continue to implement the above changing method.

Request: Implement function

expand(string s);

Where `s` is a string with the above form; return the result is a string containing only lowercase letters.

Example:

- String "2(ab3(cde)x)" changes into "abcdecdecdexabcdecdecdex".
- String "2(x0👌)3(z)" changes into "xxzzz".

Note: In this exercise, libraries `iostream`, `string` and `using namespace std;` have been used. You can add other functions for your answer, but you are not allowed to add other libraries.

For example:

Test	Result
`cout << expand("2(ab3(cde)x)") << "\n";`	`abcdecdecdexabcdecdecdex`
`cout << expand("2(x0(y))3(z)") << "\n";`	`xxzzz`

Answer: (penalty regime: 0 %)

Reset answer

```
string repeatString(string str, int times) {
    if (times <= 0) return "";
    return str + repeatString(str, times - 1);
}

string expandHelper(const string &s, int &index) {
    if (index >= static_cast<int>(s.length()) || s[index] == ')') return "";

    if (isdigit(s[index])) {
        int num = s[index] - '0';
        index++; // Move past the digit
        index++; // Move past '('

        string expandedPart = expandHelper(s, index);
        index++; // Move past ')'

        return repeatString(expandedPart, num) + expandHelper(s, index);
    }

    string currentChar(1, s[index++]);
    return currentChar + expandHelper(s, index);
}

string expand(string s) {
    int index = 0;
    return expandHelper(s, index);
}
```

	Test	Expected	Got	
✓	`cout << expand("2(ab3(cde)x)") << "\n";`	`abcdecdecdexabcdecdecdex`	`abcdecdecdexabcdecdecdex`	✓
✓	`cout << expand("2(x0(y))3(z)") << "\n";`	`xxzzz`	`xxzzz`	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **7**

Đúng

Đạt điểm 1,00 trên 1,00

Give a positive integer x, implement recursive function

```
void printHailstone(int number){}
```

to print the Hailstone Sequence of a given number upto 1 (no space at the end).

Hailstone Sequences follow these rules:

- If a number is even, divide it by 2
- If a number is odd, multiply it by 3 and add 1.

Example:

```
 If number = 5. 5 is odd number so next number is 5*3 + 1 = 16. 16 is even number so next number is 16/2
= 8...
 Finally, we get Hailstone sequence: 5 16 8 4 2 1.
```

You can find more information at: https://diendantoanhoc.net/topic/89145-d%C3%A3y-s%E1%BB%91-hailstone/

Note: Please note that you can't using key work for, while, goto (even in variable names, comment).

You can implement other recursive functions if needed.

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
printHailstone(32);	32 16 8 4 2 1

Answer: (penalty regime: 0 %)

Reset answer

```
void printHailstone(int number) {
    cout << number;
    if (number == 1) return;
    cout << " ";
    if (number % 2 == 0) {
        printHailstone(number / 2);
    } else {
        printHailstone(number * 3 + 1);
    }
}
```

	Test	Expected	Got	
✓	`printHailstone(32);`	`32 16 8 4 2 1`	`32 16 8 4 2 1`	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **8**

Đúng

Đạt điểm 1,00 trên 1,00

Function

```
int myArrayToInt(char* str, int n){}
```

takes a **string str** (which represents an positive decimal number), **n** is the number of elements in the string as arguments and returns its value.

Please note that you can't using key work for, while, goto (even in variable names, comment)

For this exercise, we have #include <iostream>, #include <string.h> and using namespace std;

For example:

Test	Result

`char str[] = "2020";`
`printf("%d", myArrayToInt(str, 4));`
	2020

Answer: (penalty regime: 0 %)

Reset answer

```
int myArrayToInt(char* str, int n) {
    if (n == 0) return 0;
    return (str[0] - '0') * pow(10, n - 1) + myArrayToInt(str + 1, n - 1);
}
```

	Test	Expected	Got	

✓	`char str[] = "2020";`
	`printf("%d", myArrayToInt(str, 4));`
		2020	2020	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **9**

Đúng

Đạt điểm 1,00 trên 1,00

Give two positive integers a and b, implement function

```
int findLCM(int a, int b){}
```

to find **LCM** (Lowest Common Multiple) of a and b using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`cout << findLCM(10, 102);`	510

Answer: (penalty regime: 0 %)

Reset answer

```
int findGCD(int a, int b) {
    if (b == 0) return a;
    return findGCD(b, a % b);
}

int findLCM(int a, int b) {
    return (a / findGCD(a, b)) * b;
}

```

	Test	Expected	Got	
✓	`cout << findLCM(10, 102);`	510	510	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **10**

Đúng

Đạt điểm 1,00 trên 1,00

Given a string `s` consisting only of `'('` and `')'`.
Your task is to implement a function with following prototype:

```
int mininumBracketAdd(string s);
```

The function returns the mininum number of brackets needed to be inserted to `s` so that the brackets are balanced.

More info:
A sequence of brackets is balanced when there are no unmatched brackets.
Example: `()(())` is balanced, but `))()` is not.

Note:
- The `iostream` library has been used and `namespace std` is being used. No other libraries are allowed.
- Using loop keywords (`for`, `while`, `do`) are not allowed, even in comments and variable names.
- You can write helper functions.

For example:

Test	Result
`cout << mininumBracketAdd(")))((");`	5

Answer: (penalty regime: 0 %)

Reset answer

```
int helper(const string &s, int index, int open, int close) {
    if (index == static_cast<int>(s.length())) return open + close;

    if (s[index] == '(')
        return helper(s, index + 1, open + 1, close);

    if (open > 0)
        return helper(s, index + 1, open - 1, close);

    return helper(s, index + 1, open, close + 1);
}

int mininumBracketAdd(string s) {
    return helper(s, 0, 0, 0);
}

```

	Test	Expected	Got	
✓	`cout << mininumBracketAdd(")))((");`	5	5	✓
✓	`cout << mininumBracketAdd("))()))()()");`	4	4	✓
✓	`cout << mininumBracketAdd("");`	0	0	✓
✓	`cout << mininumBracketAdd(")()))()))())()))))))(");`	12	12	✓
✓	`cout << mininumBracketAdd(")())((((())()())()()))))()()))(()((((()(()()()))()((())((()))))) ()((((()((((())((((()()(())(())))))(())))");`	10	10	✓

	Test	Expected	Got	
✓	cout << mininumBracketAdd(")(()())()))))((((())))))()))()))))))(()((((()(((()((()(()))())(()(()((((((()))))((((()))))(((())((()()))((())(()((((()()))()()())(())()))))(()))()))(((((()()(()))()))))()()()(()))()))()())()(()()))()))()))()))())((()()(())((()()((((()((()((((((((())()()())(()()()))()()()())()(()(())()(())((((()((((((()((())))))))()))((()()))))()()((((()()()(()())((((()(((((((())))(())))))(()))))()())())((((())()())()((((()))(())((((()((((())))((((()()(()((((()((()(()))())))))()(()(())()())((((()(()(())()()())()(()()))(((()))()())()))))((()())(())()))(()))())(()))()))()()()(()()((((((()(()()(()()))(()(()(())()(((((()((()(()(()))()())((()()((((()))(()))()(())()((()()((((((()((((()(()())())))))()((())))((()))(()())()))()())()))(()((()()((()()))(())(())((((())()())(())()((())((((()(())(()))(())()((((())((()())((()))(((((()(())()()()(()(()))(()()())()))))))())))()())(())(((((((((((()))(()))))(()))(()))(()((()()(((())))))(()(())((((()))(()))))(()((()((()(((()))()))(()((())()))))(()))(()()))()(((((()((((()()(()))()((()(((())");	70	70	✓

	Test	Expected	Got	
✓	cout << mininumBracketAdd(")()))))))()())((((()((()()))()((()()((()(((()))()))())())()))() (()()))((()()(((((()))()))()((()))()))))()(()((((((()()(()))()(())))))(()))((()(())))) (((()(())()()(((())(())()()))(((((((()()(())()()(((()(()()(()())))((()()())()(())) ()))((()())((())((())((()()()(())()()(()))((((()))()((()()))))(()((())()()())())((() (((()()()(()())(()(())(()(())()())((((()())))))(())(()(()((((()()((()(())))()))((()() (())((())))))(()))))()()(()))()((()()(((()()(()(()()()())((()))((((()(()())()((()())))() (())())(()))))))(()(())(()(()()((((((()((()((((()((()(())((()(()))()())((()()(((()()()) (()))()))(()((()))))((()()))((((()()(()((()(()())((()())())()))()()))()()())(()()))())) ()(()()()())((()))))()()()())()(()())((()()())()))())())((()((((()((()(()))()()()((() ()))(())()(()()()())((()))((()((()()()()())((()))))()())(()()))())((()(()()(((((()()() ((((((()(((()(()((((()()(()()()((((()))()()(()(()))(()()))()(()((((()()()())((((()()(())) (((()))((()()((((((()()((((()))((()((((()((((((()(()))))(((()()()(((()()((()(((((()()))))(()) ()))())(()))())))(()()(()((()))()()))))())((((((()))))))((()())()((()()()((()()(()((()) ()))()(()((())()(())))()))())))()(()()()((((((()(())()()(()))()())())()))()()))()()))((()((((())())))())((()()))((((())()))()((())()())())(()))(((()(()()()()))))(())(()((()((() ()))()(()(()((()(())(())((())((((()()()(()())()((()))((((()()()(()()))))()())()()((() ()(()())()(())()(()()(()())()())((((()(((((()(()))(()()(()))))))()))())(()())()() ((((((()((())()))))()()))()))((()))))()()))()))((()(()())()(())()()))(()()()())()()))() ()()((((((()()()()(()(())()))()()())()))(()(()))))))()))())(())()((((((()())))))()) (())((((((((()(()))))()()))((((((()(())((((()((()()((()()()())(()()))))))))()()()((((()())))) (((()))((((((()((((()()(())))()()(())))))))()(()))))()(())(())(((()))(()())))() (((()))((((()))())))(())()())(()(()((((())()())))())(()()()))))))(()(()())(()((()())()(((() (()((())(())(())))(((()(())((((((())())(()))()()()()))()((()()(())((((((())()())) (()))(()))()()))))())()()()()(()))())(()(((()(())())(()()(()()))(()()())(()()))())()) ())(())((()()()(()()(()()))(()))())(()(())((((())))))((()(((()())(())))(()()(()()(()))()() ())()()()))())(()()(()()(())())))(())))())))((((((()()((((((()))())(((((()()()))())))))) ((((()((((()(((((((((()((()(()(()())((()())()()))()(()()(()))()()(((()((((())))()((()())) ((((()()))((()((((())()()))))()()()()))))((()()()(())()(()()(()((((()())())))()))()() ()()())()))))((()(())()(()))))()())(()))()()()(((((()))(())()(())()((((((()))()))()()() ((())()())()())()(((((()()()(((()))()))))()()()()()))(()((()))))()()()(((()))()()())()()()() ()))((((((()))))))()()(())((((()))))(((()))(()))())(()(())))(((())(())))((())(())()(())() ((((())))))()()(()(()(())()()(()))())())())()(()))))((((((((()()((())())((((()()(()) ((()))))((()()()((()))((()()))()((()((()))))))))()((((()(()()()(()()))(()((((()()(()) ((((())()())()()(())()(()((())()))))()((((((()()((((()()())(()(((()(((())((()()(()))(()) ((())())))()()(()(()()(()()(())()(()))())())()((()(()(()()()))))(()()(()((())())()() ())((())()((()()()(((((()())()()(()(())()))()()))()))))((((()()((())()())((()())(((() (())((()(()(((()()()(()(()))()))()(()(()))((((()))))((()()()()(()))(())()()(()()))((()()() ()((())((()()))))(()()(())))))(())(((()()))))()()(()))))((())(()))()(()()(()())))(()) ((()()())((()(()))(()()())()((()()()))(())()))))(()()))))(()()(()()((())())()()())((())) (((()))((()(()()))((()()))()()((())((()()))))()))())()))))((()((((()(((()())()()(()))(() (()))))(()(())(((((()()(()))(()((((((()))()))(((((()((((()()((()))(((((()()())())(() ((())()()))))(())()(()()((()))))(()()))()((((((()))))((()()(()()()(((()()()))))()() (())((()()()(((((()((()()()()()(()))((((()((((()(((()()()((()()()()()((((()((((())) (()((((()((((()()()()()(())()(((()())()((((((()()))))(((()()()())()())()()(((()))))())(((())) ()(((((()(()))))))))(())(()()()))((()(((((()((())((((...snip... ()))()((()()))(()()((() ()()()(()))((((()()))()))()))(((())())))))))((((((()()(()(()()(()))()()))()())()((((((((((()((()()()())()))()(((()()(()()()))((()))((()()(()()()()()()()()()()()((()((() ()))(()()(()())(()))()()()(((())((((()((()((((()(()(()()))))(((())((()((((((()()))))() (((()))))(((()()))(()())(()))(())()(()()(()())(()))()(()))))(())(((()()()))(((()(())))) (())(((())(((((()()(()(()(())()))()(())(()((((()()()))()(())()())())(())(()()(()()((() ()))()())(())(((()()))((((()(((()()(((()()()((((()()()()((((()())(((()()((((()(())(((() ((()()))))()()()(())(((()(()()))))((()))))((((()())()()())()(()))(())(()((()(((()))) ()((((((()(((()()))))()()(())(()(()()()()(()()(()))((((((()(()()(()((((((()(((()))(())) (())((()()))()(()))))))))(())()))(())(()(()))((((()()(()((((()((((()))))(()())()()))() (()))(((()))))(()))()())(()(((()()(()())()))))())((((()((((()(()()(((()()(()))))()() ((()()((((()()))()(((((((((()()((()()(((()(())((()()()(()()(())()((()))))(())())()) ())))))((()()(((()(()((((()())()(()()(((()(()(()))()))(((()())()((()((()()(((((((()))))) ()))))()()())()(()())))))(((((()((((()))))))(()))())()()())()())))()))))()(()())())()) ()))()()()(()((()))((()())()))((((()((()(((((()((((()))()(((()())()(((((()((()()(()()(()(() (((()(((((()())(()))))(()))()(()((((()((((()(((((()(((()()(((()()()))))()()))((()(() (((()()))))()((()))))())()(()(((((()((((()(((()()(())))(()(()((((((()))(())()()))(((()((()(()(()) (())(((()(((()()(()()(()))()()))()(())))(()(()()(((()()()))(()((()()(((((())())()(((()()((() ()(((()()(((()()())))()(()()((()))()((())())))(()()())())()))((()))(()())()((((()((((()((() ()(()))))((((()))(()(((()()))((()()()))))))))()(()(((()())))))))))()(()))()()(((()() (()(()(((((()))(()((((((()(()((((((((()()()(()((((())((()))()(((()))(()((()((((()))((((() (()))()))((((((()))))))()((((()((((((()))))))()(((()()()(((())))()()))((((()()))()() (((()))(((()))()(()))))())()((()))(((((()(((()()()()())))()(((())()()(((((()))())()(()(((())))) (((((()(()((()))()))()()((()()()(((()()))((((()((()()()((()))()((()(((()()()))()))()(((()) ()))(((((())))()((()))((()()))))()))))(()((())()))()()()()((()))))()((((()(((()))))) ()(()((((((()())))))(())()((()()))(()(((((()()))(()()()))))))()()((((()(()(()()(())))))(()	82	82	✓

	Test	Expected	Got	
	((()()))(()()((())()))))))()(()()((()))()()(((()()))()))()))()()(()))))))))()((((())())()() ((((((()))()))()))((()))()(())(()())))((()))()))(()((()(()))()()()))))(()(()))))(())) ((((()(()()))))()((()))((())((())((()(())()))(()(()((()))()))(()()())()()(((((()(()((()))())) (()))(()())((())())((()()()))))()))((((())))()))))((()()((()(()((())((()()()))((()())))) ()))(()(()()(()((())())(()((()))))))(())))))()))((()))()(((()(()((((((())(()())(()()) ((())(())()))))()())(()()(())())(()()((())(()())(())(())()())((((()())()()()()))) (()))))()(())((()))((()((())()(((()(()((((()(())()()()))))()()(()))()()()())(()())))) (())(()()((()))(()())(()(()(((())))))((())((()()))()))()))((((()()())(()))()))(((((()())((((()))))()())()(()((())((()()))((())()()()(()))()))()((()()(())()))()())()))))())) ()()))()))())(())())))()))((()(())((()())(()()(()(((()(()(()))((((((())(()))()(()((()(((()())()()((()())(()(()()))((((()))()()(())(()))))(())(()(()()((((()((()(()(((((((((()))))) ()))(()(()(()((()))(()))())(())()()(()())()))()(()))()()((()(())(())())))))())()((((()))())()())()((((()())(()())((((())()()()()))((()))())())()))()))())(())()))))))(()() ())()))())()())()(()()(()()))(()))(()())((((())()()())()))())()(((()))(()(()())()(()()))()()) ())()))(()))(((((((()(())())()))(()()))))(((((()()())()))()()))(()))(())(()))()()() (((())()(()(((((()())))(()((()))(()(()()))()(()()))()((()(()())()))()()))()))(())()(()) ((())(()())()()())))())()(())((((()()()))((()()))())))))(()((())(()(()()))))))()())())) ()))))))())()(()(())(())))))((()()()))())(()(()(()())(())())((()(()(()))))(()((()()())))) ()))(((((()))))((((((())()(())(((())())");			

	Test	Expected	Got	
✓	cout << mininumBracketAdd("))()()))))())(()()))())))((()()))))(())()()(((())()())()))))))(((()) (())(())()())()()(((()))))))))((((()(((())()()(((()(()(()(()))))()(())(()()()()())())))) ()))))()))))))()())(()((()(((((()(((())))(()))))((()((())((((((()((((())))()(())()))))()) ((((((())))())(()))(())))))((()())))()())()()(()())(()))()((()((((()((((()))()())()))))) (((((((())((((())()()))()())(()()))))()))()((((((()()(((())((())(()(())(())(((()()(())() ()()))))(()))()))))(((()))()()))))(((())())(()())))())))))))())()()())()())()(((())((((())) ()(()))))()(((((()((())()((((()()(((()))(((()(((()()((()()())))))))()())()((((((())()())())))) ((((()((((()()))))()()))()(())()(((()(((()(())()())(()())())())(((((()((((((()((((()())()(() ()))()))))((((())()))()))))))(()(()))()()()))))())))))(()()))((()()()((()(())()(((())))()() ()(()(()((()()()()())(())(()))()))()))((((((()(((())))()))))(()((()))))(()))()()()((())))) ()))()())((((()))(((((()()(()()))())()((()((()(()((())()))))()))()(())))))))()))(()()))))))() ((()((((()(()()))((()))()))))((((()))((()()()()))()()(())()((()((())(())()()())(())()((((((()() ()(()))()))))))()(()))))((((())(()))))))))))((()))(()))(()()()((()(()((((()(()()()(()))() ())((((((()(()())()))))))(((((()())((((())))))()()(())(()()())))(()))()((((())()()))))() ((((())()(()))))((((())()))))(((())()()(()))()()((((()()))))())(((((((()()))))(()(((()()))(((() ()))()()((((())()))))))()(((((((())()))))))))))(()))()(())()()()(()())()()(())((((())))()(((((()) (()))))))(()))((((())())((((())()))))))()()(())(((()()(()))((((())()))))))()((((()()))()(() ((((())))(((((())))(())()))))))((((()((((((((()))(()()))())()()()(()())(((((((())())(())) ()))())(()))))))(()(((()((((())(((()))))))))()(()(()())((()(()(()))))())(()())(((())))))(() (())())))()))()((((())()(()((((()())(((()))))()(((((((()())(()))()(((())))(((((((())))(()())() ())(((())(()(())(()((((((())))()(()())(()((()())()((((())()))((()(())(((()))))())))(())((()() ()))())((((()())()(()))(())((()()(()))(())())()))(()()()))()(()))((((())))))(())()()()(())) ()((((()((((()((((((()()(())((((()))(((()((((()()())((()()))()((((((()((()())))((())() (()))()(((((()((())((())()((()((((((()))(())()(())))(()()())()(())))(())()(()))))(()()))((()))() (()((()))((()(((()))((((((()((()()((((()((()))(()(((()((()))()()))())))(((()())))()())()()) (()(()((()()()((()()))((())()()(())())))))(())()(((((()((()(())())))()((((()((((()(((()(((((((()))())))()()))))()(()()())())()((((((())))))(())()(()))(((()()((()))(((()))()((((()())) (((((((()()))))()()(()))))))(()))())())))()(()()))))))(())()(()((((((()()()()))()(()()(())) ())((()(()))())))((()((((()())(((((()((((()())(()()))(()()(()())))((()(())())(((((((((()(() ()))()())()(((())()()()())()(((((())((()))()(()))))()()))((()((((()(()())(())((((()(()()))))) (((())()))((()((()))()()))((()((((()(()))(()()))()(()))(()))((((()(())))))(((()()))((((()())) (((()((()(()))((())()))))((())))))))()()((((()((((()))))))()))((((((((()())()())()(((())()(())) ()()))()))()((((()((()))))))(()()))()))(((((()()())))(())(()))))()()(())()(()))))((((()(())() (((()))()()())((()((((()())((()))))))()())))()()()((()())()(()))()))))(())()(()))()(())()() (()))()(((()()((((())))((()))(())(()()())()))(()(()((((()(((((()))())()(()()()))))()((()))())() ((((())()))()(((((()()))((()))()()(((((())()((((()()(((()(((((()))()))(((((()()))(()(()))(() ((()()))(()))(()))(((()()((()))((((()))(())()(((((((()))))())())(((((((()(((((()))))))((())))) (()))((((((())())()()()(()())())))((((((()()((((()((((()(((()(())(()())()()(())(()()))))) (()()))(((()()))((()()()((()))())))(()()(())))))))()(())())()()((()(((((())))((()((()()(()() ((()(()((((()))))))()(()(())(()()()(((((())))()(((((()())))(((((((()((((((()()))))()))))))()() ()(()))))(()((((()())()))((((()))((((()(()))())))())))))(())(()(((((()()))()()))))(()())) (((((((()))))(())()((((((()(()()())((()))))())))(((((()()))(()())))))(()(()())((((())()(()()(()() (()))(())()(((((()((((())()))((())()))((())(()(((()(()()))()))((()((((())())())(()))()))))()(()) (()((())(()()(()))((((()(()())((())))))))))(()))()())()()))((()((((()))(((((()()))()))()))(()())))) ((()))()()(()(()(()(()(()(((()()(())())))(((())(()(()))()((((((())) ...snip...))((((())) (()()((()(()((()()()((((((()))()())(()()((((()()())()())(()(()()()())(()((((())))())(()() ((((((((()()())((()())(()))((()()((()))(()()())((()()()())(((((()()))()(((((((()))))(()) (())())(((())(()((((((())(())))((((((((()()((()(()()))()))(()()()()()))))()))((((((()))((((() ((((())()))()(()(()(((((())(()()))((()()(((()(((()())()())()()))))()()))(((((()())()(())() ((((()(()()))))((((((((((((()()()()())()(())()(())()(()))()(((()(((()(((()()))))()))(()(((()() ()()))((()))()(()())((((())())(((()())))()(()((((()()())(((()((((((()))))))(()((((()((()())) ()()))(((()))()(()()()())(((())()))((((((()(()()(((()()()()))))())))()())()()(())(((()))))()() (((()))(((())()(()))()()(((()()(()))()))(()()((((((()))()()))()))))((()(()))((()()())()))((((() (()()()()))()()))))()))()((((()))()((()))()))))()(((((())()((((()))((()(((((()(()(())()()()()) ((((((((()(()))(()))((((()()(()))()())((((()((()()()(((()(((()()))())((((()()(()((((()((((()))))(((((()()()())(((((())))(()((()))(()))()))))((((((()(()))()()())()(()()())(()((((())))) ()))()(((()))((((()())()))))))()()())((()())(()()())((()))(()((()()()(()())()()))(()()(()()(() ((()))(()))((()()((((((()()(()(()(()(()((((())))((((()()()))((()))((((((((()))()(()))(()) ()))(())()(()))()((()())()))(((()))()()()(()()())()((((()((()()(()))()(()())())((((((((())() ()))()(((((((()(()((((()())()))(((((()(()))(()))()()))()))(()()()((((()(()))()(()()((((()))(())))) ()(()))()))(((()(((())((((()((()))()(((()))()()()()((()((()(((((((((((()())()))(()))((((())() ((((()(()))))((((((()()))((((((()(((((()))()()(()(())()(()))()((((((()(()(()(()(((()(())))))) ()()))(())()))(((((()(((()(((((()()))))()()()()((((()))(()((((()(((((((((()((())(()))()(()())))) ()))((((()))(((((()((()())())((()(()))))((()(((((()((((())(()))))()((((((())))((()))()()())) ((()(()))()()((((()())(())(()()))(((())(((()()((()))()())))))))))()(()()()())(()((()))((((() ())))()(())()()()()(((()))))(()))))())))((((())(()()()))))((((())()())()())((((((()))))) ((((()))((((()(()))()())()(((())))))()()((((((()()())(()()((()(()))()()()()()()))((((()() ()()()))())(())(()(((((()((()))(((((()(((())()()(((()(()))())()(()((((())()()()))))(()) ())((((())()((()(())()(())())())(((()))())))))((()()(()))(()(()(()))())))(()()))))()()) (())))())())()()()()(((()((((())(()(()))((()(()))))())()(()(()))())(()((((()))((((((()) ()))(()()(((((())()((())(()(()((((()((())(()()(()()(())(()()))(()(()))()())(()(((())()(((((()	278	278	✓

	Test	Expected	Got	
	(()(((())))(()((()()())((((()(()()))))()()(()))))()())()))((()()(())))()))))()()()))(()((()((()((((())((((()))(())))))))))(()())()())()())()(()()(())()(()((()(())()((()(()(())(()))(()))()()()((()(()()(()()(())(()((((()()))(()((()()()))()()(())(())()()(()))))()))()(()))()(()()(((())(((())(())()()((((()))((())))((()))(()()())()()(()(())()(())()(())))))(()((())()(((()()(())(()))))())()())()(()))))(()()))((((((((())))))))()(())(((()(()(())()))(())))(())))))()))((((())(()()())()())(()))()(((()()()()(()())))(())()(((())())(())(((()(())))))(())))(()))))()))))(()))()((()))))()))(())((())()(()(()))()((()))(()((())()))((()))((()((())((()()(()))(()))))))))(((()(())()))(()((()))(((()))))()()()((()))))))())(((()())())()(()))()((((()))))()(()))((()((())))(()()))()()()))()()((()(()((()()())))))(())()))(())((((()(()((()())(())())()))))((((()(()()(()))))(())(()(())(((((())))()()(((((()(())()())(((()((())(()))()))))))()()))()())()))(())((((())(()))()(())((((())(())()()(()()(()(())()()))()()()((()))()(()))))()())()))()())((()))()(()((())(()((((())()()((()(()())))))))()))((()))(((((()())))()((()()()(((()())())(((())()))(())())(()))()()))))()))(()))(()((((((()()()))))(())()(()(()(())()))((()()()())()())())(((()(()()((((()(())()())())()())()())()()()())((()()((())((((())()())(()))((((((()()()())()(())()))()))((((()(()(()))()(()(((()()((((()()()()((((())(()()((((((()))()))()))(()))(()))(((()))(()))(()))(()(((())))))()())())()))())))()()((())(((())()(()))(())((()()())()()((((((()(()))()))()()())()(())()())(()))()))((()()(()((()()((())()()()(()(())(()(()((()()))(((())))()(()(())()(()((((())()))))()(()((()(()()(()(())))(()()))(())()))())))()");			

	Test	Expected	Got	
✓	cout << mininumBracketAdd(")()())()))()()()((((()((()()()))))())(()(()(())))))()))())((())())())()))((()()((()((((((()()())()(())))))))(()))((())((()((()))()))))()))()((()))(()))((()()()))()(())()))()))()((()()))((((()()())()())((()()))()((((((((((()(()(()(())())()))()(()()))()()))())()()))((((())(())()()()(())()((((())))))))()))))((((((((((()()))))()()))()))()())()))))))))()))(()(()))()()(()()))()))()))()))()()())(()()())(()()((((()()))()))()))()()))((()((())()()))(()))))(((((()(()()())((((())()()()(((()()()))()(()))()(())(()()))))(())))))))(((()())))))))))))()())((()())())))(((()(())(())()()()(())())))((()()())()((()))))((())()((()()())()((()()())()()(()((((()())((()())((())((()))(()()((()(()()))((()()()(()))((()()()()()))(()(()))((()(()())())))((((((()(()((((()))()()()(()()((()())()))(()()()()(()())(()))))()()((()(()((())(()))))())))())()))((((()((())(((((()()(()(())(()))()(())()(()((((()(()(())()(())(()((()))())())(()())())))(()(()))((()))((())((()(((((((()()())((((()())))()))))))(())()()(()))((()))(()()()))((((())()))))()())((())()()())((()())((())())((()))))))((()))((((())))(()))(((((())()))()()()))()(()((()))(())(())(()()()()))((()))))()))()()()()))()()()))))()))()()))((())()((())()()()))))()))))(()))))(()((((()(()()()()()()(((((()(()))(()))(((()(()()))(()))(()()(()()())()))()(())((((())(()((((()())((()((((()()))((((())(((()())()())()))()(()())()((())(()()((((()(((())())(()(())()())))()()(())(()((((()(()))()()()((((()(()))))()()((((((()(((()(())()((())()(())()((()(((()())))()))()()(()()))(()((((()(((()(((()(()((((((()((((())((()))()())((()((()(()(()))(()))()((()))((((()((()(((()()(((((()(()(((())(((((()()(((())(()(()))(())(((())(()(()())(()(()()))))(((()(((((()(())(()))()))(()))()()((()((()((()()(()()((()(()))))))))()()))((()(())()))()((()))))((()()))((()())()))()()()(()((()()(()()())))))()((((())()))()((()((((()(((())())()))((()((()))))(()((()))((((())(()()()()((()((()))(()))((((((()))()(()))(((((())))))))))))))()((()))((()))()))((()())()(()))((()((()))()()(()())((((())()))((())()()((((((((()()()())))))()))))((()((()((())((()((()())))((()))((()))()))))(()((((((()(()))((((()((()()((((()(()())))()()))))()((((()()()((()))()))(())(((())()((()))((()()(((((())(()()(()()()))()()()())))())((()))()))))((((()))((((())(()))(((()(()()())()((((((()((())(()()()()()()(((()()))()((((()()(()()(()))()))(())()()())((()(()(()))()))(()((((()((()((())(()(())((((()))()))(()((()))()((((()((()((()())()((()))()()))))(((((()()()()))()())((()())()))(((()()))(()))(()))())()()(())(()))((()((((()()))())))())))()))))(()(()((()()((()))()((((((((()()(()))))))(((())()()()(()))()())((()(()))((()(())((()()()()))()((()()))))((()()((()))))())())((()))((()()()((()()((()((()))((()()))))((((()()()()))))((()(()))()()()(()()())((((()(((()))))()))()))()((()))()))()(((()(()((((((((()))))(()(())((()((()))))))())))(()))))((()()()()()()((((()()()()))))()()))()(()))()))))()()))()()((((()()((((()(()))))((((((((())((((())((()())(())))()(((((((((())()(((((((()())()()((((((()()())((((()(())(((()()(()((())()())))))()())()))()()(())()((()))))(()((()())())(()))((())()())))(()))(()((((()))))()))))(()())())(()())())(((())()))(()()()))()(()(()()()))()(())()((()(())((((())()())()()((()(()(())))()())()(((())(()()()())()((()))))(((((()))(()()()(()))()(()))))()()())(((()()))(()())((())()((()(((()()(()(())((((()((()()((((()()(()()(((((()()((()())(()))))(((())))))(())())(((((((())()(((()()((())()((((()()))(((()(((()(((())())()((()((((((()()()(())((())))((()))(()))(()()(()()))))()))((()(((())((())()())(()))()()(((((()()))))))(((()))(()()()()()()()())))()))(()))((((()(()(()(())))(()(()))))((()))()(()()()))(()))))))(()(())(((()(()(()))()()(()))())(())())(()()(()()))(((((()()))(()(()()()(()(()()))(()()())(()))(()((((()()()))()(()()(()(())((((((((()()()))((((())((((()))(())((((((()))()()()((()(())(()))(()))))(())))))(()()))()(()())(())((()())(((((()((((()(()())))()))(()(())(()(((()(((()(((((()(())(()(((((())((((((((()()((()))))(()()(((())()())(()))))))((((()))()())((((()()()))((((()()))()) ...snip...)())((((()(()()()))((()(())))())))((()(()()()((((()))((()))(((()))))())(()(()())(((()(()()))()()(((()()()()))(()(()()(()()()))()()(())()(((()()(()))))(()(()))()))()))()))()(((()()(()))(((()())(((())(((()())(()((())))))()()))()(((()))()(())()()()))(()))))())(()))((()())(((()())()(()()))()()(()())()(()()(((())(()(()())()()()))()()(((())(())()())(((())((((((()((())()(()(((((((()()))))(((()(()))))())()(()))(()())())()(((((()()))))(()(()()))()())()()))())(()))(((())(()())(()(()(()())()(()))))()(()(((()(()))((()(()()()(((((()))())((((((()(()))(()(()()))()(()(()(((((()((()())()((((((())))))))()()(()))()))))(()(()()(()(()))((((()))))(()())(((()))())((()()(()()()(()))()(()))))))((()(())(()((()()))))()((((()(()))()()((((()(()(((()(((()(((((((()((()()(()()))()()((()()()()))()(())(((()(()()(((((()(()((((()(()))))()()())()())()()))))((()(((()())))))()())())())()())))((())())(())()(())()()())()(()()))()(()()(((()()(((()()))(()()()))(((()))())))((())))))(()(()(())(((()())))()))))((()(((()()(()(()()()))(((())))))(()(()((())(((()()()(())(())()()))(()(((((()())))()(((((()(())(()(((((()()(()(((((())))((()()(((()())()(()()()((()())))()))(()((((()())))()())(()()()(((((()()()(((((()(((()())()())()()(()((()))(()())(((((((()(((()())(((((((()(())((((()()()())()(((()((()(((()))((((((()()()())(())(()))(((()()(()))))()()()()())(((((())))))()())(()))()()(()(()(()))()((()()))())))()(())()())(()((()))()))))())()())(()()((((()(()(()((((())))))(())()((((())))(()(()(())))))))(())(((()))()())))))))(((((())((()))))()(()))()))))()()()((((()(()()())))))))()(()))))()(()))(())((((((()()()()(((())(()((((((()))(((()(((((())))()())))()))()()())))((())()(((()())()(((()(((()()(((()))()))))((())(()()()(((()))(((())()()))()()))))))((()()()(((((((()())))()()))))))(()()(())()()()(())()())()))(())((((((()(((())(((()()((((((()()()(()(()(()()()(()))()())((())))))((()()())()))())))()(()))()())()))((()((()((((())(((()))(()))()())(()))(()()())()))()(((()()(()(())((((()()(((((()))(())()((()))(()()(()))()(())((((((()()((()))()()(((((((((())(()(((()()((())(((()))()())(()))))())))()()))))((())()(()))()))))()))(())(()(((()()(()()()())()())))))(((((((((((()())(((()()()())()()(()))()(((((((()())((()))()))()())(()())()()(()(((()(()(()))()(()))))((((()))(()()()())()))((())(()))()()((((()((()())())(((()((())(((()	202	202	✓

	Test	Expected	Got	
	()()((()))())))()))))()()(()))))(()((()))))(())()()(()(())()(())((()))))()(())()))((() ((((((()(())()()(()))))))()((((()((((()())())))))((()()(()())()(((()()(())()(((()()))(((()) ()))())(()(()))))(()((((()()(((())())(()((((())(()(())))))()()(((()(())((()))(())()()))() ((())()((((()(())(((((()(())((((())(((((())))()(())((((()((((()((()))))()(())((()()()(())())))) ()(()((())()()))))()))))()()))))())())))))()())(()((((()((())()()(())(()(())((()((()())((((() ()((()()()()()()())((()))(())((((())(()))))())()())())()(()))()((()))())(())())())())()(() ((()((()()((()))))())((()()((())()((((()))())))()(((())()())((()((())(()()))))((())((((()) ()))((((()()((()))))())((()(()()))(()))((((((()))))(()())()((())(()(())(()(())(()))))())())) ((()())))()()))))()(())()(()()))()())()(()))))())(()))()())()()(()()()((()(()(())((())()) ((()((()()))((((((()()(()(()())((()((((())()())))()(()((()))((()()((()(()()(()()((()()((())((() ((((()())()())(()()())()))))((())()))))()))()()())((((()))(())()()))((((())())()()) ((())))(()))(())(()))((((())())()(())((()))()))))()())(()(()()((()()(()()(()))() ((((((())((((()(()((()))))()))))(((()()(()((((((()))()())()))((((())))))))())()((()()())()()) (()))()((()(())()((()(()(())()()(()())()((())()((((()()()((((())(()))(())(())())((((()())) ()((((()))()))))())))()())()((((()(()(()))())(())()()(()))())(()())()((()()()((()(() ((((()))()))))()((((()(()()()(((()())()()))))))()))))()((((()((()(((((()())(()))()())) ()((((((()(()()(())(()(()((())()))(()((()))(()((((())()(((())())))))((()()))((())()))(()() ((())((((())(()((((()((()(()))()()))))))))(((()()((()))()((((((()(()(()(((((()))()))))() ((((()()))((((()(()()))))))(())()((()((())())(()))((()((()))())())((())(()((()(((((");			

	Test	Expected	Got	
✓	cout << mininumBracketAdd("(()))((((())()((()()())()))()((((())(()))(((((()(())())()((((((() (()))(()()((((((()()())(()())()((((()(()(())(())(()()((()(())(())())()((((((((()())(()))) ((())()()(()()))()(()(())()))(((((()()()))()))(()()())()()))()()()()()()())((((()))))) ())((()((()((((()))()()))()()))))()())()((()()))))()())((()()(()())))()()))))((()()()(((()) ()))))()((()()((()()()((((()()()(((()()(()))()())((()()()()()()()))()()()))()()))))()()((((() ()(((())())((((()(()))))))))((()))()))(()))))))(()))))())))()(())()()(()))))()())()()((() ())((()())(()))((((()))(()))))()((()(()))()()(()()()((((())())()(()))))()))((()))))(() ()())()()))(())(()()((()()(()(()())()()()))()(()()()(()()()((()(()))))()(()(()))()))())((((() (()((()((()()(()())))(()()(()()))))(((()()(()()()()()(()))))()(()))()()()(((())(()(())())() ())(())((()()))(()()()((())()(()))()(()))))((((()()(()()(()))((((()()()((((()()))))((((())) ()))((()()))(())((()))(()))()())()))()(((()(()((()()))))(()))(()())(()))(((()()))((()((() (((()((()()((((()((()()(((()(()()((((()()())()))(())((()())())((()(((((()())(((()()))) (()))))))(())()(()(()))(((()))))((((()))(()())(()(((()))))())(()(()))(()))((((()() (((())((()))(((((()((()))(()))))))())())())(()))))(((()()))))(())((((((()((((()()())))) (()))))()()))(((()()(()())(())())()())))))))))()()())()()()()(()()()(()()(()()()()(())())) ()))())((())(())(()))(()))))())))(((((((((()(())(())()))))))())))(())(()()()(((() ((((((((((()(()))))())()()))((())(())((()))())))(((()()))))()(((())((()()()()((((((()(((() (()))))))())())())()(((()(()))()))))))))(((()()())()(()())((()()(()))()))()()((((())(()) ()(((()((()(())(()((()()())(())))(()()()))))(((()()))(((()((((((()()(()())((((()())()() ()))()(())()()()()()()((((())()()))))()())()((((()()(()(())()())()())()))))()((((((()((((() (((()()())()()))()((((())()))())(())(()())(()()(()((()((((())()())(()))()(()())()()))() ((())()(())((()((())()()()()((((((())))()())(()(())((()))(((((((()((((()()())()((((((() (((()()(())(())()))()()()))((((((()(((()()()(())(()()))()(()()()(()()()()(()())((())) ()(((()()()())()())()))))()(()()()((()))))())(((((((())(()(())))((()(((((()())(())(())())) ()()()(((())()())(((((()((()))(()))(()()()))))(())()((()((()))()()()())()()()(()(())()()((() (()(((((((())(((((()((((()()()))(()()(()))()())()(((()((((()((()()(()()(()()()()()(((((((()() ()))()()))(())()(()())()()(((())()())()())(()))()()(()))(()(((()(((()())()(())))()(()()()())(()) ())))))))))()(((()))(())(()((())(()))()))(())()()(((()(()()))((((()((())()()()())(((()))(() (()()))()()()))(()))()))))()((()))(())(()((((())))(()()))()((()())))))()()((()(())) ((((((((()())(())(((()())))())))())(((()))))(((()())))()))))((()()(()())()(())()()(((() (()))))))()((((())()())()))()(()()()())()()))()))()()(()())(())()()((((()()())()(()(()))(((() ()()((()))(((())(((()))()()))((((()(()((()(()()(()((((()((()()(()))()))()))()))(()))())() ()()())())(()))((()(()(()()())(()))(()(()))))((()(()))))()()())((()()(()()()()())(()()(()() ()((()()())((((())()))()())((()((()()()()))))))()()))((()))(()(()(()))()))()(()()(((())) ((()))()(()(()))()()))(())()))((())()()(()()((()))))()()()()())((()))))()((()(()))())) ()()())(()((((())))))(()))()((()()))())()))))((((()(()))((((())(()))(((((((()(())(())))() (()))(()))))))())()()(()(((((()(()()()()))()())()()())(((()))()())))(())(()((((())())()())() ((((()((()()))()))))((()))()))())(()(((((()()()())())())(()(((((()(()))())(()(()()) ()))))))(()))()())(()())(()((((()(()))((((()))))(())))())()())()(())(((()))(()))((()))(()) (()))()())(()()))((()(()()))(()))))))((()(()()))()((((()(()(()))()())())))))(()(((((()((...snip... ()()()((()(())()))())()))((())(((()()()()((()())())(())(()()))())())))()))))) (()()()))((()))()(())(((((()()(()()()))((()))()()((()(())(()()(()())()((())()()))))))(()) ((())()((((()((((()((()))((((()(())())()(((()())((((()((((()(()))))(()(())()()(()()(())(())()(()()) ())((()))()((()()())()))))()(())())(())(()())()((((()()(()())()(()()))()()(((()((())) ((((())))))))(())(())()))))(()))(()))(()))))()((((((()(())()()))())()(()))))((()())()()())))) ((()))))()(()((()((()))((((()(())()(()(())()())((((((((()()()(()()(())((()()))(()()()()((())) ((()()(()()((((((()))(()))(()()(()(()()(())))))((((()(((())()))((((()(())())()()(((((() (((((())(((()((()()))()))))()))((((((()()())()(((((()(())()()(()(())()(()()()()())(())) ()))()(()()((((()()))))())(((()()))))(((((()(()()()()))()(()()())(()(()()()())()(()())(())) ()(()(((()()))))()()()(())((())((()())(()()()(()((()()())()(())((((()(())())()()()))())((()()(((() (((()(((()())()))())))))()()(()(()())(()())((()()())()()()()())(((((()()))()()()((()()))))) ((()))(())()(())((()))())((((()(()()(()))((((()))()((()((((((()(()))()()())(())())())(()) ()))((()))()())((((((()(()()(())()()))))(()((((()(()((())((((((()))))(((())))))()() ())))))))(((()()()())()(((()((((()))()(()())(()(((((()()()()()))()(())((((()((()(()((((()) (())(()()(()())()(()))((()))()(()))()((((()(((((((()))))((((()(()()((((()))((()))(()) ()))((()()(())((()(()(())(())())(()))))(((())((()((((())(()))(())(())(())()((()((((()(((()) ())((()())((()))()((()))()(()))(()(()()))((((()()(()(()()))())))((((()))()((()()))()()) (((((((((((((()()())(())())())(())()))()(()()))))())(()))()(()())))(()))()()((()())()((()) ()()))))(((())(()()(()))(((()(()(()()(((((())((((((()(()()(()))())))(())(((()())))(())) ())()(()))())()((()()()))())()()(()()()()())(()()()(()()())((((((((()(()((()())())))) ()(()))((((()()()(((((()()(()()())()())(()((())(()())))))(()()(((()((((()()()))()))((()()(() (()(()))()()((()))(()(()())((()))))()()()((()))(()()()((((()(()()())()())()())()()(((() ())()()())(()))(())))))())))))()(()))())())(((()()()(((((()(())))(()(()((((((())))) ())))))))))((()))(())(())(((()(()()()(((()(()(()))((()()))((((()()))())()(()()))(()) ()(())((((())())((()())(((((()(()(()()))(()))(())(())(((()((((()())((()()))(((())))() (())()()(()))(()()))))()()(()))))))(()))((()()((())))(()))))))()()))))())((()(()))()	426	426	✓

	Test	Expected	Got	
	()))(())))()())((()((())()(()()))()((())()))(())()(()()))))()()))))()()()()))()(((())) ())(((((((()))((()()()())())((()))()()())()))())(()()()(()((((())()()()))))(()()()()))))) ()))()())((()(((()((()))(((()(((()((())()))))())((())()(()(()(((()()))(())(())((())()) (()))()()(()((((())(())())((()()()(()((()(()((()))()(()))()())()((((((())()(())(())) ((((())(()))(()(()((()((()(((((())))((())(())((()))(()))()()(()))(()(())(()))()())(() ()))(()))()()()(()())(((()))((()))(())(()))((()())()()((()(()))())))(())((((()))())) ())))))))))()(()())((((()((()((()(()()))()()(()(())(()()(()()(()((((()()(()()(()()())) ((((()))()()(()()()((()))((((((()))())(()((()(()))))()))()(()()((()())(()(((()((()((() ((())()()))))()(()))()(())()(()((())()))(()())((()(())(())((()(()(()(())())(()()(() (())())(()((((())(()(()))(()))(()(()()())((()()(()))(()(())())()))))()((()(((()))()() (())()(()((()))()(()))((((((()))())(())((()((((()((((()))())))()))()()((()((()(((()()())))) ())((()()(())(()))()(()))()(()(())())()))(()()()))()())(()))))(()()(()))(()((()))()) ((((()))())(()()(()()()))((((((())(()))((()((()))(()(()))))(()))))()(()))))())((())()))()) ()((()))))())(()))()())()((())((())((()))))))))()(((()((()(()()(()()(()))())()()((()) ((())()))((())((()((()(()())(()()((()(())()(())(((((()()()(((()()()()()())(((((() ((((()(()))((())(()(())((())((()())((())((())(())((())((()((()))(()))))(())(())())(())) (())(((((()()(()))())(()))((()((()((()(())((()((()()))()()())()(()())((()(((()()(()())) ()))))())(()())((())))))))))((()()(()((())(()((()((()((()()))))()(()))())((()(()((()() ())())((()))(()()((()())(()))(()()(()))()((()()((()))()))))))))))((())()))(()(()()) (");			

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **11**

Đúng

Đạt điểm 1,00 trên 1,00

Given a string `s` representing a sentence consisting only of `a`-`z` and `A`-`Z` and space character.
Your task is to implement a function with following prototype:

```
string reverseSentence(string s);
```

The function returns the reverse sentence of sentence `s`.
The testcases ensure that there is only one space character between two adjacent words, and the sentences do not begin or end with any space characters.

Note:
- The `iostream` library has been used and `namespace std` is being used. No other libraries are allowed.
- Using loop keywords (`for`, `while`, `do`) are not allowed, even in comments and variable names.
- You can write helper functions.

For example:

Test	Result
`cout << reverseSentence("data structure and algorithm is scary");`	`scary is algorithm and structure data`

Answer: (penalty regime: 0, 0, 0, 5, 10, 15, ... %)

Reset answer

```
string extractWord(const string &s, int start, int end) {
    if (end == static_cast<int>(s.length()) || s[end] == ' ')
        return s.substr(start, end - start);
    return extractWord(s, start, end + 1);
}

int findNextStart(const string &s, int end) {
    if (end == static_cast<int>(s.length())) return end;
    return s[end] == ' ' ? end + 1 : findNextStart(s, end + 1);
}

string helper(const string &s, int start) {
    if (start >= static_cast<int>(s.length())) return "";

    string word = extractWord(s, start, start);
    int nextStart = findNextStart(s, start + word.length());

    string remaining = helper(s, nextStart);
    return remaining.empty() ? word : remaining + " " + word;
}

string reverseSentence(string s) {
    return helper(s, 0);
}


```

	Test	Expected	Got	
✓	`cout << reverseSentence("data structure and algorithm is scary");`	`scary is algorithm and structure data`	`scary is algorithm and structure data`	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **12**

Đúng

Đạt điểm 1,00 trên 1,00

Given a string, implement function

```
int strLen(char* str){}
```

to calculate length of the string using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result

`char str[] = "Truong DH Bach Khoa";`
`cout << strLen(str);`
	19

Answer: (penalty regime: 0 %)

Reset answer

```
int strLen(char* str) {
    if (*str == '\0') return 0;
    return 1 + strLen(str + 1);
}

```

	Test	Expected	Got	

✓	`char str[] = "Truong DH Bach Khoa";`
	`cout << strLen(str);`
		19	19	✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.