

Примерни тестове
за устен изпит по Увод в програмирането
(време за работа 60 мин.)
Част 1

Трите теста по-долу са давани на студентите от сп. Информатика на редовната и на септемврийската сесия.

Подобни задачи могат да се намерят в сборника по УП.

Увод в програмирането на базата на C++, вариант 1

17.02.2011

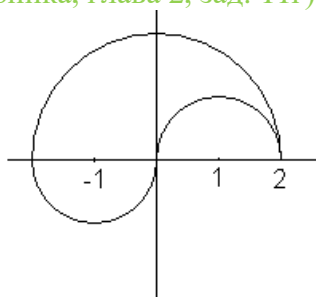
Задача 1 (4 точки). Да се запише булев израз, който има стойност истина, ако посоченото условие е вярно и стойност – лъжа, в противен случай:

а) поне едно от числата a , b и c е положително; `bool x = a>0||b>0||c>0`

б) булевата променлива x има стойност *true*, а булевата променлива y – стойност *false*;

`bool b = x&&!y`

в) точка принадлежи във вътрешността и контура на очертаната фигура: *има я дословно в сборника, глава 2, зад. 11г)*



д) цифрите на трицифреното естествено число x образуват растяща редица.

`bool b = x/100 >= x/10%10 && x/10%10 >= x%10`

Задача 2 (3 точки). Кое условие е в сила след ключовата дума *else* на условния оператор:

а) `if (a == -3 || a == 0 || b == 3 || b == 10) b = 5; else b = 10;`

`a!=-3 && a!=0 && b!=3 && b!=10`

б) `if (a < 5 || (a > 10 && a < 20)) b = a; else a = b;`

`a>=5 && (a<=10 || a>=20)`

в) `if (a <= 1 || (a >= 3 && !(a == 5))) a = 1; else a = 0;`

`a>1 && (a<3 || a==5)`

Задача 3 (4 точки). Да се напише условен оператор, който е еквивалентен на оператора за присвояване `x = a&&b || c || d;` където всички променливи са булеви и в който не се използват логически операции.

`bool x = false;`

`if(a){`

```

    if(b) {
        x = true;
    } else {
        x = false;
    }
} else if(c) {
    x = true;
} else if (d) {
    x = true;
} else {
    x = false;
}

```

Задача 4 (2 точки). Да се напише оператор *switch*, който по дадено цяло число от 1 до 12, задаващо пореден номер на месец, извежда в кой сезон от годината е този месец (да се счита, че месеците 1, 2 и 3 са зима; 4, 5 и 6 – пролет; 7, 8 и 9 – лято и 10, 11 и 12 - есен).

```

switch(month) {
    case 1:
    case 2:
    case 3: cout << "winter"; break;
    case 4:
    case 5:
    case 6: cout << "spring"; break;
    case 7:
    case 8:
    case 9: cout<< "summer"; break;
    case 10:
    case 11:
    case 12: cout << "autumn"; break;
    default : cout << "wrong input!";
}

```

Задача 5 (3 точки). Нека n дадено цяло число ($n > 1$). Да се напише програмен фрагмент, който намира стойността на израза:

$$\sqrt{1 + \sqrt{4 + \sqrt{7 + \dots \sqrt{(n-3) + \sqrt{n}}}}}$$

```

int result = 0;
int n;
cin >> n;
for (int i = n; i >= 1; i -= 3) {
    result += sqrt(i);
}

```

Задача 6 (4 точки). Нека x и y са цели числа, като $y \geq 0$. Да се заменят „.....” в програмата по-долу така, че получената програма да реализира степенуването x^y по следната формула:

$$x^y = \begin{cases} 1, & y = 0 \\ (x^2)^{\frac{y}{2}}, & y > 0, y - \text{четно} \\ x \cdot x^{y-1}, & y > 0, y - \text{нечетно} \end{cases}$$

```
#include <iostream>
using namespace std;

int main()
{ int x = 0, y = 0, pwr = 1;
  cout << "Въведете цяло число: ";
  cin >> x;
  cout << "Въведете неотр. цяло число: ";
  cin >> y;
  while (y > ..... ) // 0
  { if (y % 2 == 0)
    { x = .....; // x*x
      y = .....; // y / 2
    }
    else
    { y = .....; // y - 1
      pwr = .....; // x*pow(x, y)
    }
  }
  cout << "power=" << pwr << "\n";
  return 0;
}
```

Задача 7 (4 точки). Да се напише програмен фрагмент, който проверява дали в редицата от числа a_0, a_1, \dots, a_{n-1} има елемент, който принадлежи на интервала $[p, q]$.

```
int n;
cout << "enter size: ";
cin >> n;
int arr[n];
for(int i = 0; i < n; i++){
    cin>>arr[i];
}
int p,q;
cout << "enter interval.";
cin>>p>>q;
bool fl = 0;
for(int i = 0; i < n; i++){
    if(arr[i]>=p&&arr[i]<=q){
        fl = 1;
        break;
    }
}
```

```

}
if(fl) {
    cout << "yes";
} else {
    cout << "no";
}

```

Задача 8 (7 точки). Да се напише булева функция, която проверява дали редица от $3n$ цели числа има вида: $b_1, b_2, \dots, b_n, b_1, b_2, \dots, b_n, b_1, b_2, \dots, b_n$. За целта могат да се дефинират и използват помощни функции

```

int n;
    cout << "enter size: ";
    cin >> n;
    int arr[n];
    for(int i = 0; i < n; i++)
    {

        cin>>arr[i];
    }
    int m = n/3;
    bool fl = 1;
    for(int i = 0; i < n-m; i++)
    {

        if(arr[i]!=arr[i+m])
        {

            fl = 0;
            break;
        }
    }
    cout << fl;

```

Задача 9 (5 точки). Какъв е резултатът от изпълнението на програмата?

<pre> #include <iostream> using namespace std; void func(int &x, int *y, int z) { int a = 7; int b = 3; x = z + b; *y = z + x; z = a; cout << "func: x = " << x << endl; cout << "func: y = " << *y << endl; cout << "func: z = " << z << endl; cout << "func: a = " << a << endl; cout << "func: b = " << b << endl; return; } </pre>	<pre> func: x = 4 func: y = 5 func: z = 7 func: a = 7 func: b = 3 main: a = 5 main: b = 4 </pre>
--	--

<pre> int main() { int a = 5; int b = 1; func(b, &a, b); cout << "main: a = " << a << endl; cout << "main: b = " << b << endl; return 0; } </pre>	
---	--

Задача 10 (7 точки). Да се дефинира функция, която извежда на екрана елементите на квадратна матрица като ги обхожда по диагонали, успоредни на главния диагонал, започвайки от горния десен ъгъл.

```

int m[4][4] = { {1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16} };
int n = 4;
for (int offset = -n + 1; offset < n; ++offset) {
    for (int i = 0; i < n - abs(offset); ++i) {
        if (offset <= 0) {
            cout << m[i][i - offset] << " ";
        } else {
            cout << m[i + offset][i] << " ";
        }
    }
    cout << endl;
}

```

Задача 11 (6 точки). Да се дефинира **рекурсивна** функция, която намира броя на срещанията на дадена цифра k в десетичния запис на дадено естествено число n .

```

int meet(int n, int k){
    if (k>9) {
        return 0;
    }
    if(n<10) {
        return n==k;
    }
    return meet(n/10,k)+(n%10==k);
}

```

Задача 12 (7 точки). Да се дефинира **рекурсивна** функция, която проверява дали елементите на редица са различни.

```

bool diff(int* arr, int n){
    if(n==1) {
        return true;
    }
    return arr[0] != arr[1] && diff(arr+1, n-1);
}

```