

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

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

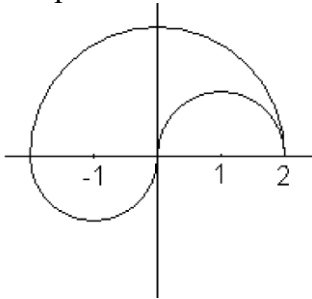
*Подобни задачи могат да се намерят в сборника по УП.*

**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;
- б) булевата променлива  $x$  има стойност *true*, а булевата променлива  $y$  – стойност *false*;
- в) точка принадлежи във вътрешността и контура на очертаната фигура:



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

**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;`
- б) `if (a < 5 || (a > 10 && a < 20)) b = a; else a = b;`
- в) `if (a <= 1 || (a >= 3 && !(a == 5))) a = 1; else a = 0;`

**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.

**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).

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

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

**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 > ..... )
  { if (y % 2 == 0)
    { x = .....;
      y = .....;
    }
    else
    { y = .....;
      pwr = .....;
    }
  }
  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  $[p, q]$ .

**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

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

|   |  |
|---|--|
| <pre>#include &lt;iostream&gt; using namespace std; void func(int &amp;x, int *y, int z) { int a = 7;   int b = 3;   x = z + b;   *y = z + x;   z = a;   cout &lt;&lt; "func: x = " &lt;&lt; x &lt;&lt; endl;   cout &lt;&lt; "func: y = " &lt;&lt; *y &lt;&lt; endl;   cout &lt;&lt; "func: z = " &lt;&lt; z &lt;&lt; endl;   cout &lt;&lt; "func: a = " &lt;&lt; a &lt;&lt; endl;   cout &lt;&lt; "func: b = " &lt;&lt; b &lt;&lt; endl;   return; } int main() { int a = 5;   int b = 1;   func(b, &amp;a, b);   cout &lt;&lt; "main: a = " &lt;&lt; a &lt;&lt; endl;   cout &lt;&lt; "main: b = " &lt;&lt; b &lt;&lt; endl;   return 0; }</pre> |  |
|---|--|

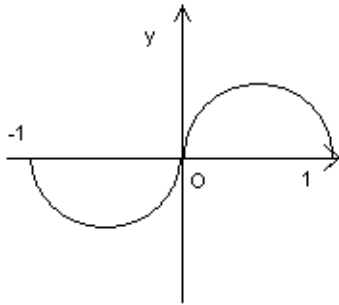
**Задача 10 (7 точки).** Да се дефинира функция, която извежда на екрана елементите на квадратна матрица като ги обхожда по диагонали, успоредни на главния диагонал, започвайки от долния ляв ъгъл.

**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  $n$ .

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

**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) числата  $a$ ,  $b$  и  $c$  са различни;
- б) поне две от цифрите на трицифреното число  $m$  са равни помежду си;
- в) точка принадлежи на поне един от полукръговете (включително контура им);



- г) цифрата 5 участва в записа на положителното трицифрено число  $q$ ;

**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;`
- б) `if ((a > -5 && a < 0) || a > 2) b = a; else a = b;`
- в) `if (!(a <= 1) && a < 3 || !(a <= 5)) a = 1; else a = 0;`

**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.

**Задача 4 (2 точки).** Да се напише оператор `switch`, която по дадено цяло число от 5 до 20, извежда дали е четно или е нечетно. В случай, че числото е нечетно да извежда и дали числото е просто или не е просто.

**Задача 5 (3 точки).** Да се напише програмен фрагмент, който намира стойността на израза:  $\sin(a + \sin(a + 2 + \sin(a + 4 + \dots \sin(b - 2 + \sin(b)) \dots)))$ , където  $a$  и  $b$  са дадени цели числа.

**Задача 6 (4 точки).** Да се заменят „.....” в програмата по-долу така, че получената програма да проверява дали сумата от цифрите на дадено естествено число е кратна на 3.

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

int main()
{ cout << "Въведете естествено число: ";
  int n; cin >> n;
  // намиране на сумата от цифрите на n
  int s = 0;
  int k = n;
  while (k != ..... )
  { s = .....;
    k = .....
  }
  // проверка дали сумата от цифрите се дели на 3
  if (s ..... 3 == ..... ) cout << "Сумата от цифрите на " << n
                                << " се дели на 3. \n";
  else cout << "Сумата от цифрите на " << n
            << " не се дели на 3. \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 не принадлежи на интервала  $[p, q]$ .

**Задача 8 (7 точки).** Да се напише булева функция, която проверява дали редица от  $2n$  цели числа има вида:  $b_1, b_2, \dots, b_n, b_1+1, b_2+2, \dots, b_n+n$ . За целта могат да се дефинират и използват помощни функции.

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

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

void func(int &x, int y, int* z)
{ int a = 3;
  int b = 1;
  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;
}

int main()
{ int a = 2;
  int b = 4;
  func(b, a, &b);
  cout << "main: a = " << a << endl;
  cout << "main: b = " << b << endl;
  return 0;
}
```

**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.

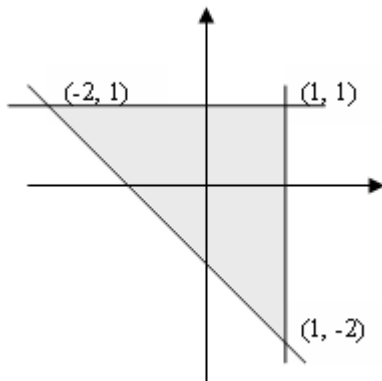
**Задача 11 (6 точки).** Да се дефинира **рекурсивна** функция, която проверява дали дадено естествено число е степен на 2.

**Задача 12 (7 точки).** Да се дефинира **рекурсивна** функция, която проверява дали елементите на редица са трион, т.е. имат вида  $a_0 > a_1 < a_2 > a_3 < \dots$



**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) числата  $a$ ,  $b$ ,  $c$  и  $d$  образуват монотонно растяща редица;
- б) булевата променлива  $x$  има стойност *false*, а булевата променлива  $y$  – стойност *true*;
- в) точка лежи извън очертаната фигура:



- д) цифрите на четирицифреното естествено число  $x$  са различни.

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

- а) `if (a == -5 || a == -1 || b == 2 || b == 9) b = 1; else b = 5;`
- б) `if (a < 0 || (a >= 3 && a <= 8)) b = -a; else a = -b;`
- в) `if ((a > 3 || !(a == 5)) && a <= -5) x = 1; else x = 0;`

**Задача 3 (4 точки).** Да се напише условен оператор, в който не се използват логически операции, който е еквивалентен на оператора за присвояване `b1 = x && y || z;` (всички променливи са булеви).

**Задача 4 (2 точки).** Да се напише оператор *switch*, който по дадено цяло число от 1 до 7, задаващо пореден ден на седмицата, извежда дали денят е работен или почивен.

**Задача 5 (3 точки).** Да се напише програмен фрагмент, който намира стойността на израза:  $\cos(a + \cos(a^2 + \cos(a^3 + \dots + \cos(a^{b-1} + \cos(a^b)) \dots)))$ , където  $a$  и  $b$  са дадени цели числа.

**Задача 6 (4 точки).** Да се заменят „...” в програмата по-долу така, че получената програма да проверява дали сумата от цифрите на дадено естествено число е кратна на 5.

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

int main()
{ cout << "Въведете естествено число: ";
  int n; cin >> n;
  // намиране на сумата от цифрите на n
  int s = 0;
  int k = n;
  while (k != ..... )
  { s = .....;
    k = .....
  }
  if (s ..... 5 == ..... ) cout << "Сумата от цифрите на " << n
                                << " се дели на 5. \n";
  else cout << "Сумата от цифрите на " << n
            << " не се дели на 5. \n";
  return 0;
}
```

**Задача 7 (4 точки).** Да се напише програмен фрагмент, който проверява дали в редицата от цели числа  $a_0, a_1, \dots, a_{n-1}$  има елемент, който е степен на 5 (степените на 5 са:  $1, 5, 5^2, 5^3, 5^4, \dots$ ).

**Задача 8 (7 точки).** Да се напише булева функция, която проверява дали редица от  $2n$  цели числа има вида:  $b_1, b_2, \dots, b_{n-1}, b_n, b_n, b_{n-1}, \dots, b_2, b_1$ .

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

|   |  |
|---|--|
| <pre>#include &lt;iostream&gt; using namespace std; void func(int &amp;x, int *y, int z) { int a = 1;   int b = a+3;   x = z + b;   *y = z * x;   z = a*b;   cout &lt;&lt; "func: x = " &lt;&lt; x &lt;&lt; endl;   cout &lt;&lt; "func: y = " &lt;&lt; *y &lt;&lt; endl;   cout &lt;&lt; "func: z = " &lt;&lt; z &lt;&lt; endl;   cout &lt;&lt; "func: a = " &lt;&lt; a &lt;&lt; endl;   cout &lt;&lt; "func: b = " &lt;&lt; b &lt;&lt; endl;   return; } int main() { int a = 3;   int b = 2;   func(b, &amp;a, b);   cout &lt;&lt; "main: a = " &lt;&lt; a &lt;&lt; endl;   cout &lt;&lt; "main: b = " &lt;&lt; b &lt;&lt; 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 вторичния главен диагонал, започвайки от горния ляв ъгъл.

**Задача 11 (6 точки).** Да се дефинира **рекурсивна** функция, която намира сумата от цифрите в десетичния запис на дадено естествено число  $n$ .

**Задача 12 (7 точки).** Да се дефинира **рекурсивна** функция, която проверява дали елементите на числова редица са числа на Фибоначи. *Пример.* Редицата 5, 13, 1, 0, 34, 1, 0, 55, 2, 3 удовлетворява свойството.

*Забележка.* 0 и 1 са числа на Фибоначи; всяко следващо число на Фибоначи е сума на предшестващите го две числа на Фибоначи, т.е. числата на Фибоначи принадлежат на редицата 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...