Образец оформления отчёта по Лабораторной работе.

|  |  |
| --- | --- |
| **К Г Э У** | МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

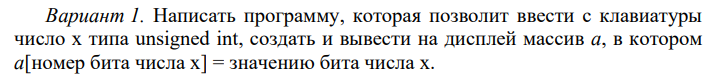
**Кафедра Информатики и информационных управляющих систем**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8**

**РЕШЕНИЕ ЗАДАЧ С ОДНОМЕРНЫМИ СТАТИЧЕСКИМИ МАССИВАМИ МЕТОДАМИ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ. АНАЛИЗ КОДОВ ПРОГРАММ С ПЕРЕДАЧЕЙ ПАРАМЕТРОВ.**

|  |  |
| --- | --- |
| **Исполнитель:** | Соловьёв Леонид |
| **Группа:** | ПИ-1-22 |
|  |  |
|  |  |

Казань – 2022

1.1  
  
Код  
#include <iostream>

const int DIM = 32;

short int mass[DIM];

void init\_arr( unsigned &x, short int mass[]);

void cout\_arr( short int mass[]);

int main() {

    unsigned x = 0;

    std::cout << "Enter x: "; std::cin >> x;

    std::cout << "x = " << x << std::endl;

    init\_arr(x, mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned &x, short int mass[]) {

    unsigned n = x;

    unsigned i = 0;

    while (n != 0) {

        mass[DIM - i - 1] = n % 2;

        n /= 2;

        i++;

    }

}

void cout\_arr(short int mass[]) {

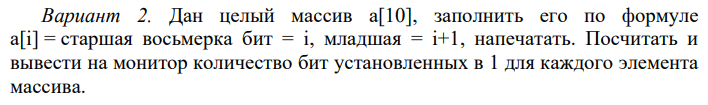
    for (int i = 0; i < DIM; i++) {

        std::cout << mass[i];

    }

}  
Решение  


1.2

  
Код  
#include <iostream>

#include <cmath>

#include <ctime>

#include <string>

const int DIM = 10;

short int mass[DIM];

short int mass\_count\_one[DIM];

std::string mass\_binary\_num[DIM];

void init\_arr(int DIM, short int mass[]);

void count\_ones(int DIM, short int arr\_num[], short int arr\_counts[]);

void binary\_num(int DIM, short int arr\_num[], std::string arr\_binary\_num[]);

void prntRes(int DIM, short int arr\_num[], std::string arr\_binary\_num[], short int arr\_counts[]);

int main() {

    init\_arr(DIM, mass);

    count\_ones(DIM, mass, mass\_count\_one);

    binary\_num(DIM, mass, mass\_binary\_num);

    prntRes(DIM, mass, mass\_binary\_num, mass\_count\_one);

    return 0;

}

void init\_arr(int DIM, short int mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = (i << 8) | (i+1);

    }

}

void count\_ones(int DIM, short int arr\_num[], short int arr\_counts[]) {

    for (int i = 0; i < DIM; i++) {

        unsigned int count = 0;

        int n = arr\_num[i];

        while (n != 0) {

            count += n%2;

            n >>= 1;

        }

        arr\_counts[i] = count;

    }

}

void binary\_num(int DIM, short int arr\_num[], std::string arr\_binary\_num[]) {

    for (int i = 0; i < DIM; i++) {

        std::string S;

        int n = arr\_num[i];

        while (n > 0) {

            S = std::string(1, (char) (n%2 + 48)) + S;

            n /= 2;

        }

        arr\_binary\_num[i] = S;

    }

}

void prntRes(int DIM, short int arr\_num[], std::string arr\_binary\_num[], short int arr\_counts[]) {

    for (int i = 0 ; i < DIM; i++) {

        std::cout << i+1 << ". num = " << arr\_num[i] << "\t bin\_num = " << arr\_binary\_num[i] << "\t count of 1 = " << arr\_counts[i] << std::endl;

    }

}

//REAL CASE, BEST PRACTICE

/\*

#include <iostream>

#include <cmath>

#include <ctime>

#include <string>

const int DIM = 10;

short int mass[DIM];

short int mass\_count\_one[DIM];

int main() {

    for (int i = 0; i < DIM; i++) {

        unsigned short int count = 0;

        mass[i] = (i << 8) | (i+1);

        short int n = mass[i];

        while (n != 0) {

            count += n%2;

            n >>= 1;

        }

        std::cout << mass[i] << '\t';

        n = mass[i];

        std::string S;

        while (n>0) {

            S = std::string(1, (char) (n%2 + 48)) + S;

            n /= 2;

        }

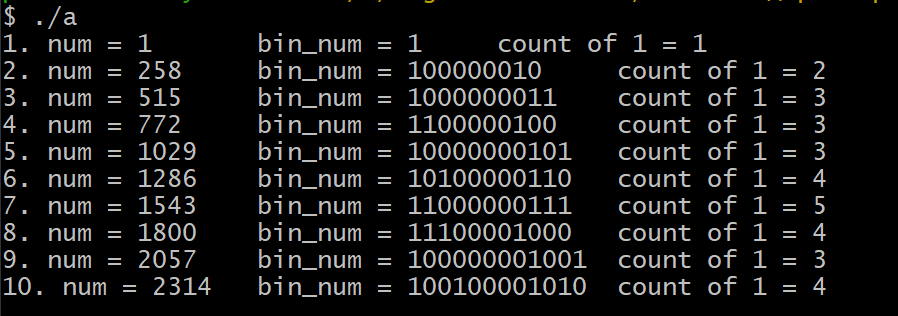
        std::cout << S << '\t' << count << std::endl;

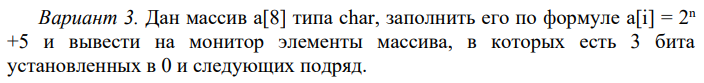
    }

}

\*/

Решение



1.3  
  
Код  
#include <iostream>

#include <cmath>

const int DIM = 8;

char mass[DIM];

void init\_arr(char mass[]);

void cout\_arr(char mass[]);

bool binary\_num\_valid(unsigned x);

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(char mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = std::pow(2, i) + 5;

    }

}

void cout\_arr(char mass[]) {

    for (int i = 0; i < DIM; i++) {

        if (binary\_num\_valid((unsigned int) mass[i])) {

            std::cout << i+1 << ". " << (int) mass[i] << '\t' << binary\_num((int) mass[i]) << std::endl;

        }

    }

}

bool binary\_num\_valid(unsigned x) {

    unsigned count\_zero = 0;

    bool flag = false;

    while (x > 0) {

        if (x % 2 == 0) {

            count\_zero++;

        } else {

            count\_zero = 0;

        }

        if (count\_zero == 3) {

            flag = true;

        }

        x /= 2;

    }

    return flag;

}

std::string binary\_num(unsigned x) {

    std::string S;

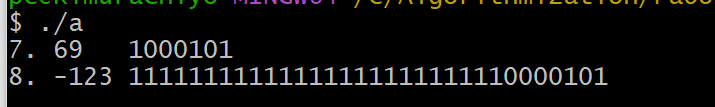
    while (x > 0) {

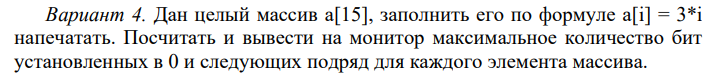
        S = std::string(1, (char) (x%2 + 48)) + S;

        x /= 2;

    }

    return S;

}  
Решение  


1.4  
Код

#include <iostream>

#include <cmath>

const int DIM = 15;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_zero(unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

unsigned all\_max\_count\_zero = 0;

unsigned max\_count\_zero\_num;

unsigned max\_count\_zero\_num\_index;

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    std::cout << "max zeros in num = " << all\_max\_count\_zero << "\nthis num = " << max\_count\_zero\_num;

    std::cout << "\nhis index in list(higher) = " << max\_count\_zero\_num\_index + 1;

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = 3 \* i;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        std::cout << i+1 << ". " << mass[i] << '\t' << binary\_num(mass[i]) << '\t' << binary\_max\_count\_zero(mass[i], i) << std::endl;

    }

}

unsigned binary\_max\_count\_zero(unsigned x, unsigned ind) {

    unsigned num = x;

    unsigned max\_count\_zero = 0;

    unsigned count\_zero = 0;

    while (x > 0) {

        if (x % 2 == 0) {

            count\_zero++;

        } else {

            if (count\_zero > max\_count\_zero) {

                max\_count\_zero = count\_zero;

            }

            count\_zero = 0;

        }

        x /= 2;

    }

    if (max\_count\_zero > all\_max\_count\_zero) {

        all\_max\_count\_zero = max\_count\_zero;

        max\_count\_zero\_num = num;

        max\_count\_zero\_num\_index = ind;

    }

    return max\_count\_zero;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

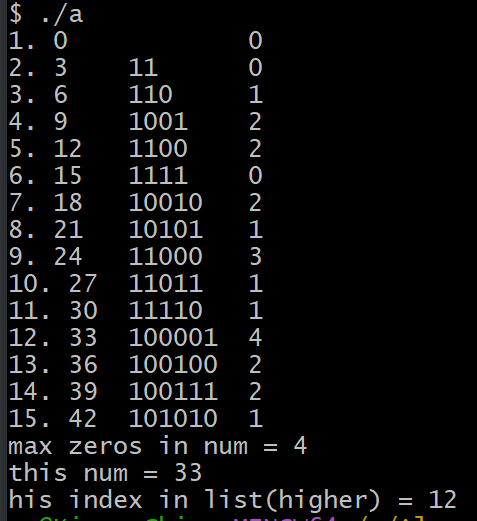
        S = std::string(1, (char) (x%2 + 48)) + S;

        x /= 2;

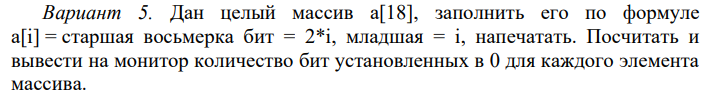
    }

    return S;

}

Решение  


1.5



Код  
#include <iostream>

#include <cmath>

const int DIM = 18;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned count(unsigned x);

unsigned count\_zero = 0;

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = ((2 \* i) << 23) | i;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        std::cout << i+1 << ". " << mass[i] << '\t' << binary\_num(mass[i]) << '\t' << count(mass[i]) << std::endl;

    }

}

unsigned count(unsigned x) {

    count\_zero = 32;

    while (x > 0) {

        if (x % 2 == 1) {

            count\_zero--;

        }

        x /= 2;

    }

    return count\_zero;

}

std::string binary\_num(unsigned x) {

    std::string S;

    if (x == 0) {

        return std::string(32, '0');

    }

    unsigned num\_of\_bytes = 31;

    while (num\_of\_bytes > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

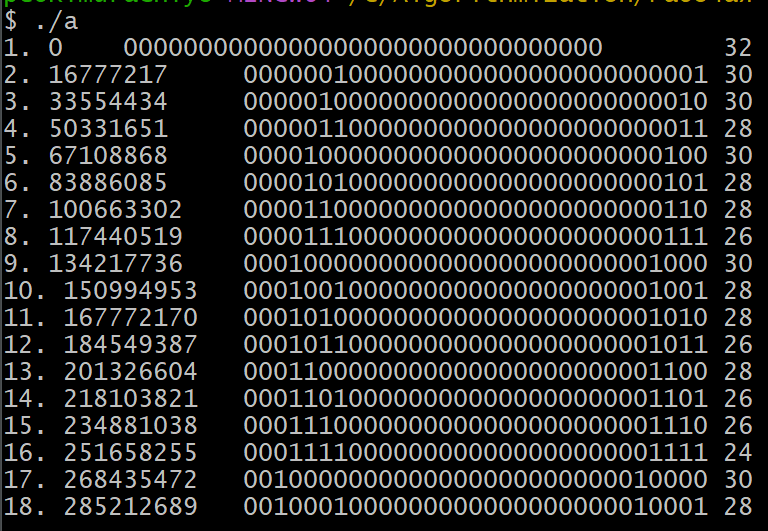
        x /= 2;

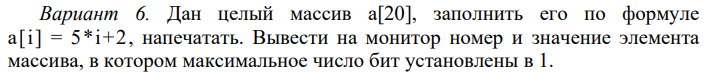
        num\_of\_bytes--;

    }

    return S;

}  
Решение



1.6  
  
Код

#include <iostream>

#include <cmath>

const int DIM = 15;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

unsigned all\_max\_count\_one = 0;

unsigned max\_count\_one\_num;

unsigned max\_count\_one\_num\_index;

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    std::cout << "max zeros in num = " << all\_max\_count\_one << "\nthis num = " << max\_count\_one\_num;

    std::cout << "\nhis index in list(higher) = " << max\_count\_one\_num\_index + 1;

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = 3 \* i;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        std::cout << i+1 << ". " << mass[i] << '\t' << binary\_num(mass[i]) << '\t' << binary\_max\_count\_one(mass[i], i) << std::endl;

    }

}

unsigned binary\_max\_count\_one(unsigned x, unsigned ind) {

    unsigned num = x;

    unsigned count\_one = 0;

    while (x > 0) {

        if (x % 2 == 1) {

            count\_one++;

        }

        x /= 2;

    }

    if (count\_one >= all\_max\_count\_one) {

        all\_max\_count\_one = count\_one;

        max\_count\_one\_num = num;

        max\_count\_one\_num\_index = ind;

    }

    return count\_one;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

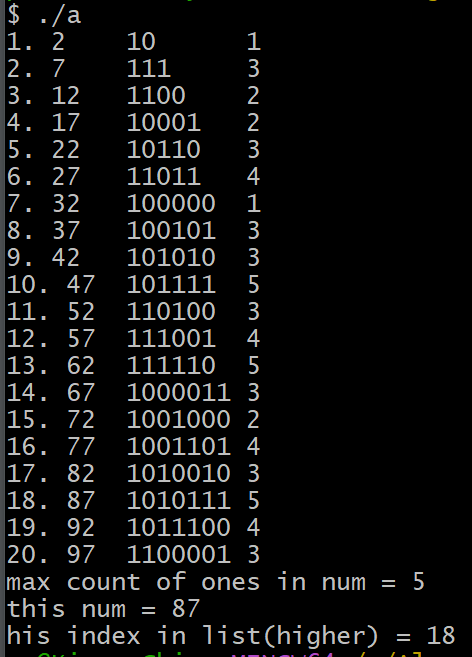
        x /= 2;

    }

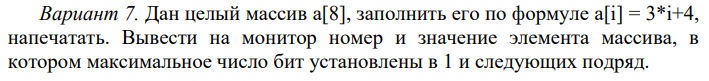
    return S;

}

Решение



1.7

  
Код

#include <iostream>

#include <cmath>

const int DIM = 8;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

unsigned all\_max\_count\_one = 0;

unsigned max\_count\_one\_num = 0;

unsigned max\_count\_one\_num\_index = 0;

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    std::cout << "max ones in num = " << all\_max\_count\_one << "\nthis num = " << max\_count\_one\_num;

    std::cout << "\nhis index in list(higher) = " << max\_count\_one\_num\_index + 1;

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = 3 \* i + 4;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        std::cout << i+1 << ". " << mass[i] << '\t' << binary\_num(mass[i]) << '\t' << binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i) << std::endl;

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0, max\_count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        std::cout << \*it;

        if (ch == '1') {

            count++;

        } else {

            count = 0;

        }

        if (count > max\_count) {

            max\_count = count;

        }

        if (max\_count > all\_max\_count\_one) {

            all\_max\_count\_one = max\_count;

            max\_count\_one\_num = x;

            max\_count\_one\_num\_index = ind;

        }

    }

    std::cout << "  ";

    return max\_count;

}

std::string binary\_num(unsigned x) {

    std::string S;

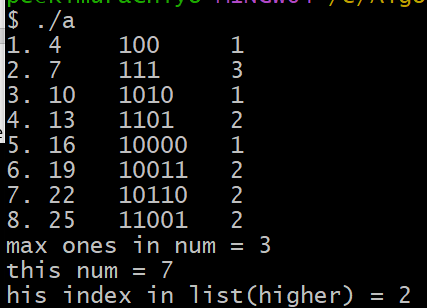
    while (x > 0) {

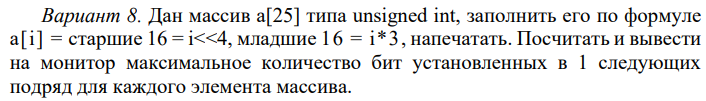
        S = std::string(1, (char) (x%2 + 48)) + S;

        x /= 2;

    }

    return S;

}  
Решение  


1.8  
  
Код

#include <iostream>

#include <cmath>

const int DIM = 25;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

unsigned all\_max\_count\_one = 0;

unsigned max\_count\_one\_num = 0;

unsigned max\_count\_one\_num\_index = 0;

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = ((i\*8) << 16) | (i \* 3);

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        std::cout << i+1 << ". " << mass[i] << '\t' << binary\_num(mass[i]) << "\t    " << binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)<< std::endl;

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        if (ch == '1') {

            count++;

        }

    }

    return count;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

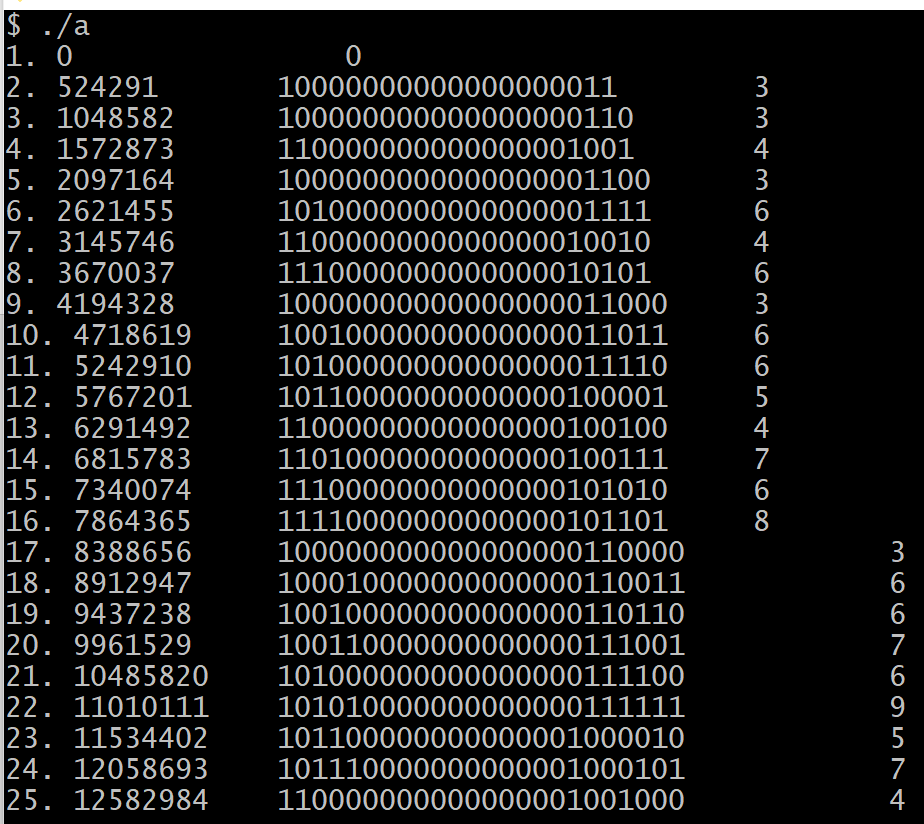
        x /= 2;

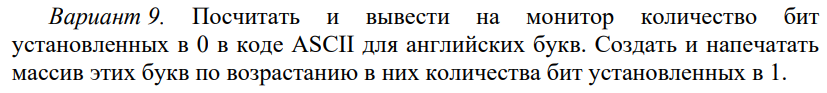
    }

    return S;

}

Решение



1.9  
  
Код  
#include <iostream>

#include <cmath>

#include <string>

const int DIM = 26;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = i + 97;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0, spec\_num = 0; i < DIM; i++, spec\_num++) {

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

            if (spec\_num ==  binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)) {

                std::cout << i+1 << ". " << (char) mass[i] << '\t' << binary\_num(mass[i]) << "\t    " << binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)<< std::endl;

            }

        }

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        if (ch == '1') {

            count++;

        }

    }

    return count;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

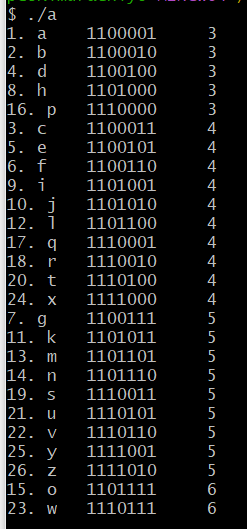
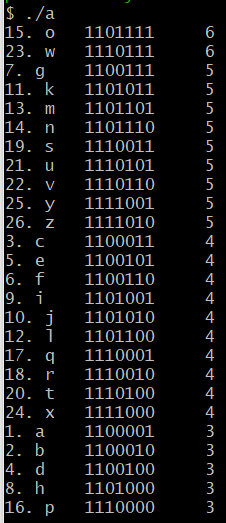
        x /= 2;

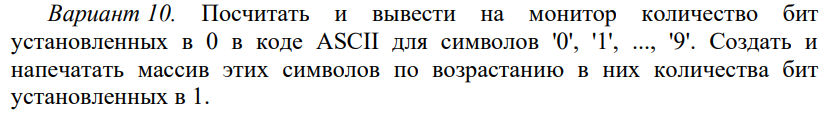
    }

    return S;

}

Решение



1.10  
  
Код

#include <iostream>

#include <cmath>

#include <string>

const int DIM = 10;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = i + 48;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0, spec\_num = 0; i < DIM; i++, spec\_num++) {

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

            if (spec\_num ==  binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)) {

                std::cout << i+1 << ". " << (char) mass[i] << '\t' << binary\_num(mass[i]) << "\t    " << binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)<< std::endl;

            }

        }

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        if (ch == '1') {

            count++;

        }

    }

    return count;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

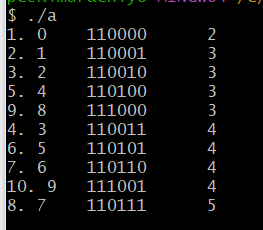
        x /= 2;

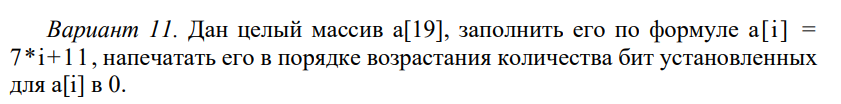
    }

    return S;

}

Решение



1.11  
  
Код  
#include <iostream>

#include <cmath>

#include <string>

const int DIM = 19;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = 7\*i + 11;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0, spec\_num = 0; i < DIM; i++, spec\_num++) {

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

            if (spec\_num ==  binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)) {

                std::cout << i+1 << ". " <<  mass[i] << '\t' << binary\_num(mass[i]) << "\t    " << binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)<< std::endl;

            }

        }

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        if (ch == '0') {

            count++;

        }

    }

    return count;

}

std::string binary\_num(unsigned x) {

    std::string S;

    while (x > 0) {

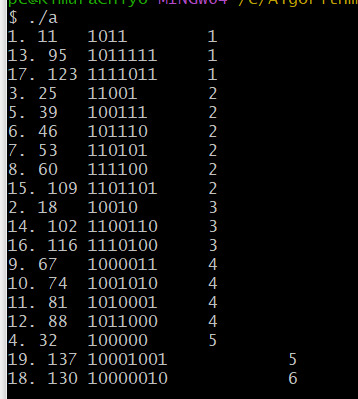
        S = std::string(1, (char) (x%2 + 48)) + S;

        x /= 2;

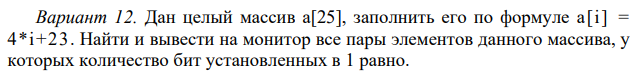
    }

    return S;

}  
Решение



1.12



Код

#include <iostream>

#include <cmath>

#include <string>

const int DIM = 19;

unsigned mass[DIM];

void init\_arr(unsigned mass[]);

void cout\_arr(unsigned mass[]);

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind);

std::string binary\_num(unsigned x);

int main() {

    init\_arr(mass);

    cout\_arr(mass);

    return 0;

}

void init\_arr(unsigned mass[]) {

    for (int i = 0; i < DIM; i++) {

        mass[i] = 4\*i + 23;

    }

}

void cout\_arr(unsigned mass[]) {

    for (int i = 0, spec\_num = 0; i < DIM, spec\_num < 10; i++, spec\_num++) {

        if (spec\_num < 3 || spec\_num > 6) {

            continue;

        }

        std::cout << "count of ones = " << spec\_num << ": ";

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

            if (spec\_num ==  binary\_max\_count\_one(binary\_num(mass[i]),mass[i], i)) {

                std::cout << mass[i] << "(" << binary\_num(mass[i]) << ")" << "\t";

            }

        }

        std::cout << '\n';

    }

}

unsigned binary\_max\_count\_one(std::string S, unsigned x, unsigned ind) {

    std::string n = S;

    std::string::iterator it = S.end();

    unsigned count = 0;

    for (std::string::iterator it = S.begin(); it < S.end(); it++) {

        char ch = \*it;

        if (ch == '1') {

            count++;

        }

    }

    return count;

}

std::string binary\_num(unsigned x) {

    std::string S;

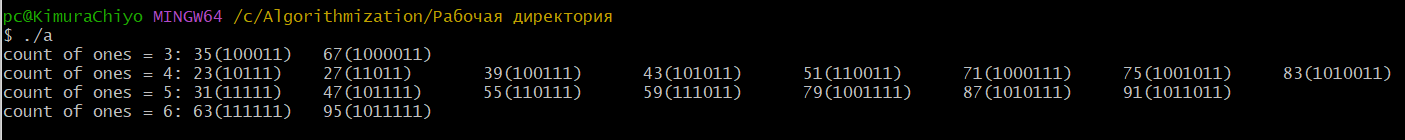
    while (x > 0) {

        S = std::string(1, (char) (x%2 + 48)) + S;

        x /= 2;

    }

    return S;

}  
Решение  


2.1

1)Объявление и инициализация pi(float):pi = atan(1)\*4 = 0.785398163 \* 4 = 3.14159265

2)Объявление и инициализация массива с(float elements), в котором 6 элементов  
0: 0.523598776, 1: 0.785398163, 2: 1.04719755, 3: 1.57079633,

4: 2.0943951, 5: 3.14159265

3)Размер массива n: n = 4 \* 6 / 4 = 6

4)Объявление и инициализация s(float): s = 1/1000 \* 1000 = 1.0

5)Цикл по четным элементам: 0 2 4

0:s = s + cos(элемент с индексом 1) + sin(элемент с индексом 0) =   
1 + 0.7071067 + 0.5 = 2.2071067

2:s = s + cos(элемент с индексом 3) + sin(элемент с индексом 2) =

= 2.2071067 + 0 + 0.8660254038 = 3.073132104

4:s = s + cos(элемент с индексом 5) + sin(элемент с индексом 4) =

3.073132104 + (-1) + 0.8660254038 = 2.939157508

6)С новой строки: s = 2.939157508

2.2

1)Объявление и инициализация tn, tk, dt: tn = -3, tk = 3, dt = 2

2)Объявление и инициализация n = 2

3)Вывод с новой строки: “tab”t”tab”y”переход на новую строку через одну”

4)Цикл: -3 -1 1 3  
-3:Вывод:”tab”-3”tab”prog(-3, 2)(-29.5)”переход на новую строку”

prog(-3, 2):   
a)Объявление и инициализация fi = 20, u = 1  
б)Цикл: 1 2

1: u = 9, fi = 20 – 9/1 = 11  
2: u = 81, fi = 11 – 81/2 = 11 – 40.5 = -29.5

-1:Вывод:”tab”-1”tab”prog(-1, 2)(18.5)”переход на новую строку”

prog(-1, 2):  
a) Объявление и инициализация fi = 20, u = 1  
б)Цикл: 1 2  
1: u = 1, fi = 20 – 1/1 = 19  
2: u = 1, fi = 19 – 1/2 = 18.5

1:Вывод:”tab”1”tab”prog(1, 2)(18.5)”переход на новую строку”

prog(1, 2):  
a) Объявление и инициализация fi = 20, u = 1  
б)Цикл: 1 2  
1: u = 1, fi = 20 – 1/1 = 19  
2: u = 1, fi = 19 – 1/2 = 18.5

3:Вывод:”tab”3”tab”prog(3, 2)(-29.5)”переход на новую строку”

prog(3, 2):  
a) Объявление и инициализация fi = 20, u = 1  
б)Цикл: 1 2  
1: u = 9, fi = 20 – 9/1 = 11  
2: u = 81, fi = 11 – 81/2 = 11 – 40.5 = -29.5

2.3

1)Объявление и инициализация zn, zk, dz: zn = -10, zk = 11, dz = 5

2)Вывод с новой строки: “tab”t”tab”y”переход на новую строку через одну”

3)Цикл: -10 -5 0 5 10

-10: Вывод:”tab”-10”tab”bond(-10)(-1)”переход на новую строку”

bond(-10):

1) Объявление и инициализация а: a = 312.7

2)-10 < 0 -> (int) t%3 = (int) -10%3 = -1  
-5: Вывод:”tab”-5”tab”bond(-5)(-2)”переход на новую строку”

bond(-5):

1) Объявление и инициализация а: a = 312.7

2)-10 < 0 -> (int) t%3 = (int) -5%3 = -2  
0: Вывод:”tab”0”tab”bond(0)(1)”переход на новую строку”

bond(0):  
1) Объявление и инициализация а: a = 312.7

2)0 !< 0 && 0 !> 0 -> cos(51\*t) -> cos(0) -> 1  
5: Вывод:”tab”5”tab”bond(5)(2.7)”переход на новую строку”

bond(5):

1) Объявление и инициализация а: a = 312.7

2)5 > 0 -> fmod(312.7, 5) -> 2.7  
10: Вывод:”tab”10”tab”bond(10)”переход на новую строку”

bond(10):

1 Объявление и инициализация ммма: a = 312.7

2)10 > 0 -> fmod(312.7, 10) -> 2.7

2.4  
1)Объявление и инициализация a, b, c: a = -11, b = 7, c = 6

2)Пропуск строки и вывод: “ Construction table of selected function””переход на следующую строку”

3)constr(a, b, c)  
constr(-11, 7, 6):  
Цикл: -11 -5 1 7  
-11: Вывод:”tab”-11”tab”res(-11, 5)(-2)”переход на новую строку”

res(-11, 5):  
1)Инициализация z  
2)-11 < 0 -> z = x/st -> z = -11/5; -> z = -2

-5: Вывод:”tab”-5”tab”res(-5, 5)(-1)”переход на новую строку”

res(-5, 5):  
1)Инициализация z  
2)-5 < 0 -> z = x/st -> z = -5/5; -> z = -1

1: Вывод:”tab”1”tab”res(1, 5)(0.2)”переход на новую строку”

res(1, 5):  
1)Инициализация z

2)1 > 0 -> z = (float)x / st -> 0.2   
7: Вывод:”tab”7”tab”res(7, 5)(1.4)”переход на новую строку”

res(7, 5):  
1)Инициализация z

2)7 > 0 -> z = (float)x / st -> 1.4

5)переход на новую строку через одну

6)вывод: sizeof(2) -> 4

2.5

1)Объявление x, eps, z:  
2)Ввод x: x = 0.1  
3)Ввод eps через строку: eps = 0.05  
4)z = d(0.1, 0.05) = 0.98

d(0.1, 0.05):

a)Объявление и инициализация sum, u, cf: sum = 1, u = 2, cf = -1

b)Цикл: 1

1:u = u \* x \* x = 2 \* 0.1 \* 0.1 = 0.02

sum = sum + cf\*u = 1 + (-1)\*0.02 = 0.98  
 cf = -cf = -(-1) = 1  
 Вывод c новой строки: “ i = 1”В

5)Вывод с новой строки: “For x = 0.1 z(x) = 0.98””переход на новую строку”

2.6  
1)Объявление и инициализация tn, tk, dt: tn = -9, tk = 9, dt = 3  
2)Вывод с новой строки: “tab”t”tab”y”переход на новую строку через одну”

3)Цикл: -9 -6 -3 0 3 6 9(10)

-9: Вывод: “tab”-9”tab”z(-9)”переход на новую строку”  
z(-9):

1)-9 < 0 -> 81  
-6: Вывод: “tab”-6”tab”z(-6)”переход на новую строку”  
z(-6):

1)-6 < 0 -> 36

-3: Вывод: “tab”-3”tab”z(-3)”переход на новую строку”  
z(-3):

1)-3 < 0 -> 9  
0: Вывод: “tab”0”tab”z(0)”переход на новую строку”  
z(0):

1)0 !< 0 && 0 !> 0 -> 1  
3: Вывод: “tab”3”tab”z(3)”переход на новую строку”  
z(3):

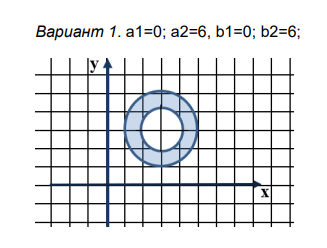
1) 3 > 0 -> fmod(10, 3) -> 1  
6: Вывод: “tab”6”tab”z(6)”переход на новую строку”  
z(6):

1) 6 > 0 -> fmod(10, 6) -> 4

9: Вывод: “tab”9”tab”z(9)”переход на новую строку”

z(9):

1. 9 > 0 -> fmod(10, 9) -> 1

3.1  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

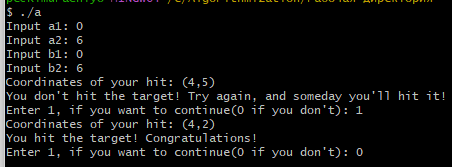
    y = rand()%((b2 - a1) + 1) + b1;

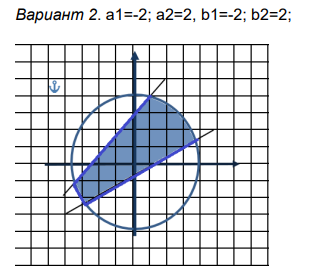
}

bool validationCoordinates(int x, int y) {

    return ((pow((x - 3), 2) + pow((y - 3), 2) > pow(1, 2)) && (pow((x - 3), 2) + pow((y - 3), 2) < pow(2, 2)));

}  
Решение



3.2  


Код

#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

    y = rand()%((b2 - a1) + 1) + b1;

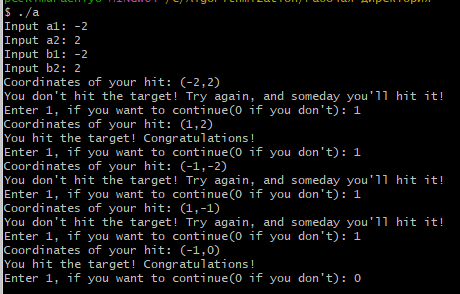
}

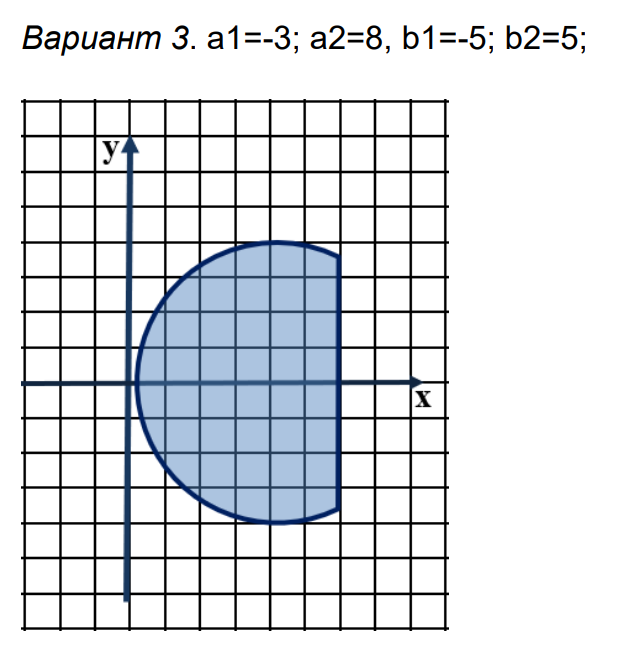
bool validationCoordinates(int x, int y) {

    return ((pow(x, 2) + pow(y, 2) < pow(4, 2)) && (y <= 1.2\*x + 3) && (y >= (4./7)\*x - (5./7)));

}

Решение



3.3  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

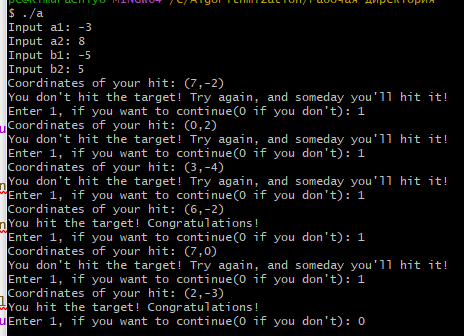
    x = rand()%((a2 - a1) + 1) + a1;

    y = rand()%((b2 - a1) + 1) + b1;

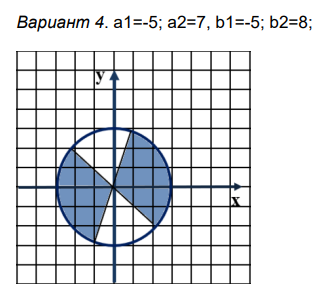
}

bool validationCoordinates(int x, int y) {

    return ((pow((x - 4), 2) + pow(y, 2) <= pow(4, 2)) && (x <= 6));

}  
Решение  


3.4

  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

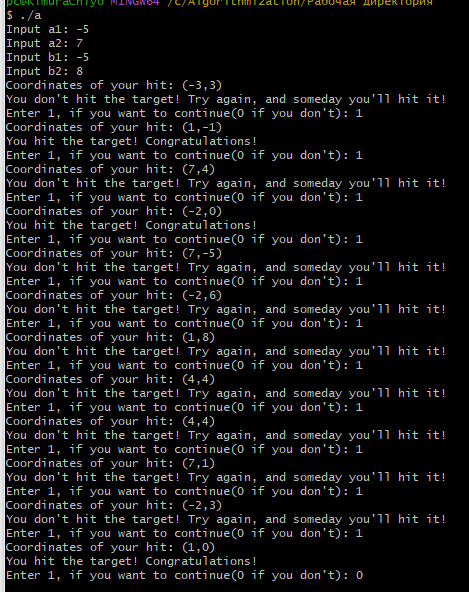
    y = rand()%((b2 - a1) + 1) + b1;

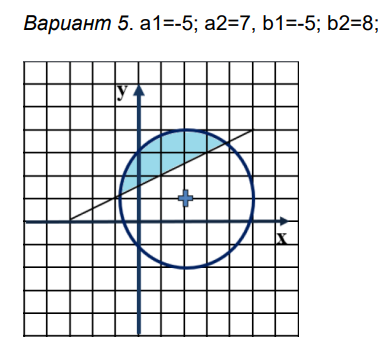
}

bool validationCoordinates(int x, int y) {

    return (x < 0) ? (pow(x, 2) + pow(y, 2) <= pow(3, 2) && (y >= 3\*x) && (y <= -x)) : (pow(x, 2) + pow(y, 2) <= pow(3, 2) && (y <= 3\*x) && (y >= -x));

}

Решение  


3.5  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

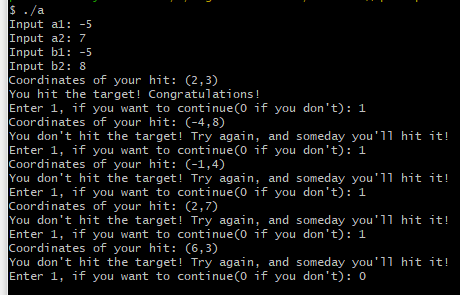
    y = rand()%((b2 - a1) + 1) + b1;

}

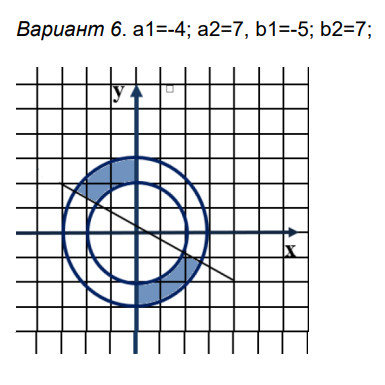
bool validationCoordinates(int x, int y) {

    return ((y >= 0.5\*x + 1.5) && (pow((x - 2), 2) + pow((y - 1), 2) <= pow(3, 2)));

}

Решение  


3.6

  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

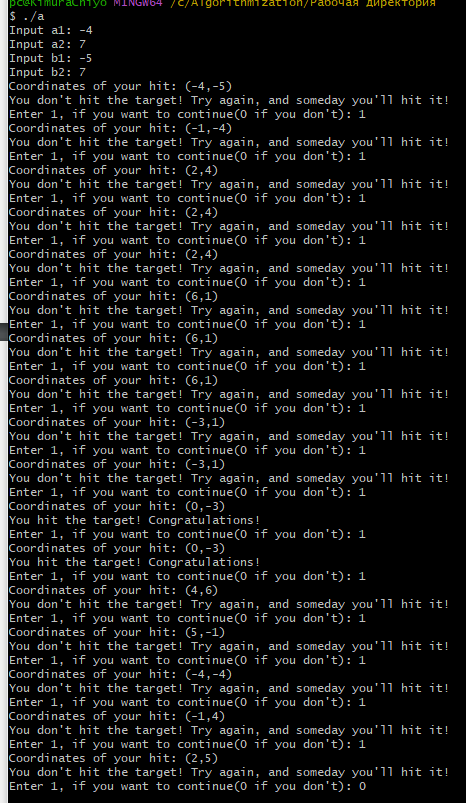
    srand(time(NULL));

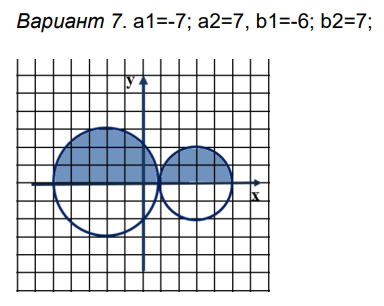
    x = rand()%((a2 - a1) + 1) + a1;

    y = rand()%((b2 - a1) + 1) + b1;

}

bool validationCoordinates(int x, int y) {

    return (x < 0) ? (pow(x, 2) + pow(y, 2) <= pow(3, 2) && (pow(x, 2) + pow(y, 2) >= pow(2, 2)) && (y >= (-1./2)\*x)) : (pow(x, 2) + pow(y, 2) <= pow(3, 2) && (pow(x, 2) + pow(y, 2) >= pow(2, 2)) && (y <= (-1./2)\*x));  
}  
Решение  


3.7  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

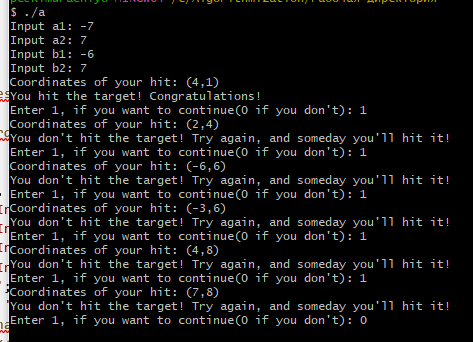
    y = rand()%((b2 - a1) + 1) + b1;

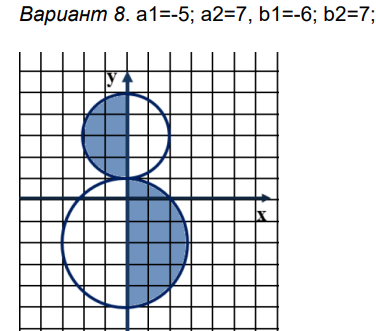
}

bool validationCoordinates(int x, int y) {

    return ((y >= 0) && ((pow(x + 2, 2) + pow(y, 2) <= pow(3, 2)) || (pow(x - 3, 2) + pow(y, 2) <= pow(2, 2))));

}  
Решение



3.8  
  
Код

#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

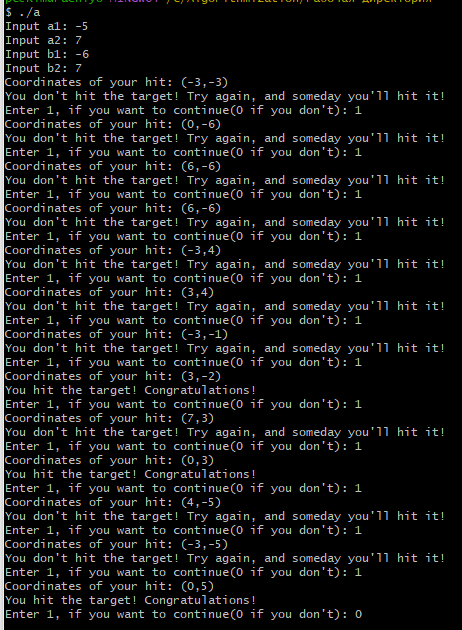
    y = rand()%((b2 - a1) + 1) + b1;

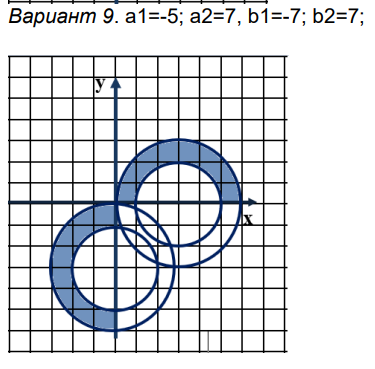
}

bool validationCoordinates(int x, int y) {

    return (y > 1) ? ((x <= 0) && (pow(x, 2) + pow((y - 3), 2) <= pow(2, 2))) : ((x >= 0) && (pow(x, 2) + pow((y + 2), 2) <= pow(3, 2)));

}  
Решение



3.9  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

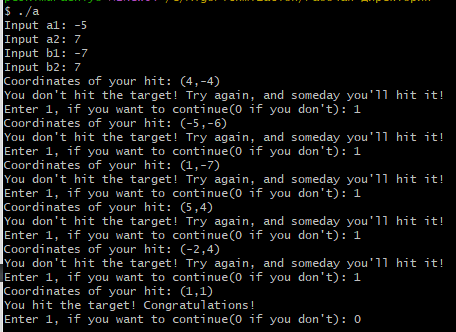
    y = rand()%((b2 - a1) + 1) + b1;

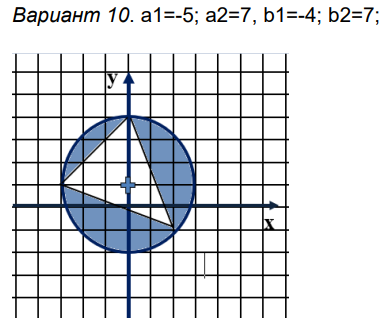
}

bool validationCoordinates(int x, int y) {

    return (((y >= 0) && (pow(x-3, 2) + pow(y, 2) >= pow(2, 2)) && (pow(x-3, 2) + pow(y, 2) <= pow(3, 2))) || ((y < 0) && (pow(x, 2) + pow(y-3, 2) >= pow(2, 2)) && (pow(x, 2) + pow(y-3, 2) <= pow(3, 2))));

}  
Решение



3.10  
  
Код  
#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

    y = rand()%((b2 - a1) + 1) + b1;

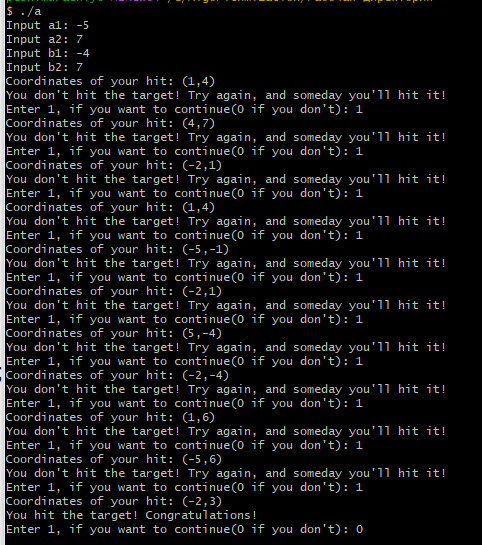
}

bool validationCoordinates(int x, int y) {

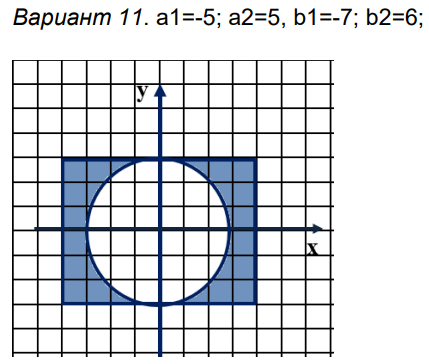
    return ((pow(x, 2) + pow(y-1, 2) <= pow(3, 2)) && ((y >= x + 4) || (y >= (-5./2)\*x + 4) || (y <= (-2./5)\*x - (1./5))));

}

Решение



3.11

  
Код

#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

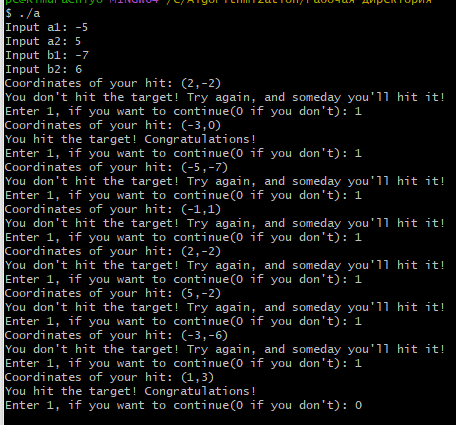
    y = rand()%((b2 - a1) + 1) + b1;

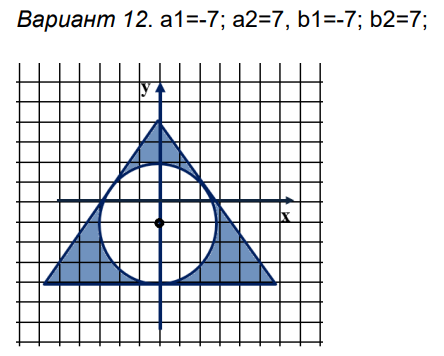
}

bool validationCoordinates(int x, int y) {

    return ((x >= -4 && x <= 4) && (y >= -3 && y <= 3) && (pow(x, 2) + pow(y, 2) >= pow(3, 2)));

}  
Решение



3.12  
  
Код

#include<iostream>

#include <cmath>

#include <ctime>

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y);

bool validationCoordinates(int x, int y);

int main() {

    int a1, a2, b1, b2, x, y;

    std::cout << "Input a1: "; std::cin >> a1;

    std::cout << "Input a2: "; std::cin >> a2;

    std::cout << "Input b1: "; std::cin >> b1;

    std::cout << "Input b2: "; std::cin >> b2;

    char flag = '1';

    while (flag != '0') {

        randCoordinates(a1, a2, b1, b2, x, y);

        std::cout << "Coordinates of your hit: (" << x << "," << y << ")" << std::endl;

        if (validationCoordinates(x, y)) {

            std::cout << "You hit the target! Congratulations!" << std::endl;

        } else {

            std::cout << "You don't hit the target! Try again, and someday you'll hit it!" << std::endl;

        }

        std::cout << "Enter 1, if you want to continue(0 if you don't): "; std::cin >> flag;

    }

    return 0;

}

void randCoordinates(int a1, int a2, int b1, int b2, int &x, int &y) {

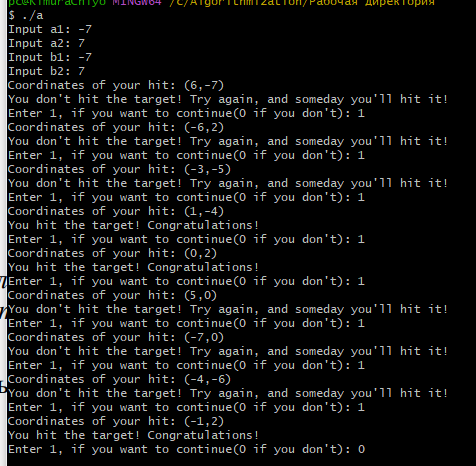
    srand(time(NULL));

    x = rand()%((a2 - a1) + 1) + a1;

    y = rand()%((b2 - a1) + 1) + b1;

}

bool validationCoordinates(int x, int y) {

    return ((pow(x, 2) + pow(y + 1, 2) >= pow(3, 2)) && (y <= (-3./2)\*x + 4) && (y <= (3./2)\*x + 4) && (y >= -4));  
}  
Решение  


4.1

Программа выведет abcde

1)с0: a, c1: b, c2: c, c3: d, c4: e, c5: \0

b0: o, b1: p, b2: r, b3: \0  
2)Вызов функции f(a, b)

а) Сопоставляем фактические параметры с формальными  
 указатель а указывает на первый элемент а – передача по значению

указатель b указывает на первый элемент b – передача по значению

b) Присвоение а = b реализуется на копию первого элемента а, поэтому значение первого элемента а не изменится

Присвоение b = a реализуется на копию первого элемента b, поэтому значение первого элемента b не изменится

с) Возврат в main  
3)Вывод:abcde  
  
4.2

1)a0: 6, a1: -1, a2: 0, a3: 3, a4: 2  
2)Вызов функции f(a[0], a[4])  
 a) Сопоставляем фактические параметры с формальными  
 ячейка а копирует значение а[0]  
 ячейка b принимает ссылку на а[4]  
 b) Присвоение а = 7 реализуется на копию первого элемента а, поэтому значение первого элемента а не изменяется  
 Присвоение d = -1 реализуется по ссылке на a[4], поэтому значение а[4] изменится на -1  
 с) Возврат в main  
3)Вывод:6 0 -1

4.3

1)a0: г, a1: о, a2: л, a3: о, a4: с, a5: \0  
2)Вызов функции f(a[0], a[4])  
 a) Сопоставляем фактические параметры с формальными  
 ячейка a принимает ссылку на а[0]  
 ячейка b принимает ссылку на а[4]  
 b) Присвоение а = ‘в’ реализуется по ссылке на а[0], поэтому  
значение а[0] изменится на ‘в’

Присвоение d = ‘c’ реализуется по ссылке на a[4], поэтому значение а[4] не изменится, потому что было ‘c’ и до присвоения  
 с) Возврат в main  
3)Вывод:волос

4.4  
1)Объявление a,b,c,d  
2)Инициализация a = 1, c = 0, b = 3  
3)Вызов функции f(b, a, c)  
 a) Сопоставляем фактические параметры с формальными  
 ячейка а копирует значение b  
 ячейка с принимает ссылку на а  
 ячейка d принимает ссылку на с  
 b) Присвоение а = 5 реализуется по копии b -> значение b не изменится  
 Присвоение с = 7 реализуется по ссылке на а -> значение а изменится на 7  
 Присвоение b = 9 реализуется для локальной для функции переменной b, значение b в main не изменится  
 с) Возврат в main  
4)Вывод:703

4.5

1)a0: 2, a1: -1, a2: 3, a3: 0  
 b0: 4, b1: -1, b2: 0  
2)Вызов функции f(a[2], b[1])

а) Сопоставляем фактические параметры с формальными  
 ячейка а принимает ссылку на а[2]

ячейка b принимает ссылку на b[1]

b) Присвоение а = b реализуется по ссылке на а[2] -> значение а[2] изменится на значение b[1]  
 Присвоение а = b реализуется по ссылке на b[1] -> значение b[1] изменится на значение a[2](которое стало значением b[1])

с) Возврат в main

4)Вывод:-1 -1 -1 0

4.6

4.7

4.8

R = X + Y; X = min(X, Y); Y = R – X;

if (X > Y) {X = X + Y; Y = X – Y; X = X – Y;}

R = (X < Y) ? X : Y; Y = X + Y – R; X = R;

if (X > Y) R = X; else R = Y; X = X + Y – R; Y = R;

R = X \* Y; if (X > Y) X = Y; Y = R / X;  
Это действительно так, я проверил  
4.9  
12456

4.10

123 – подходят  
4 – не подходит, потому что &a принимает только переменную

4.11

34 – подходят  
1 – не подходит, с – указатель на символ, а должен быть float

2 – не подходит, потому что 3 – число, а должна быть переменная int

4.12

Ответ: 2, потому что сначала указатель а указывал на 0 элемент массива, а после инкремента он стал указывать на 1 элемент массива -> значение по указателю стало равно b[1], которое равно 10

ДЗ1

1)Объявление и инициализация xn, xk, dx: xn = -8, xk = 8, dx = 4;

2)Вывод:”переход на следующую строку”” Construction table of selected function””переход на следующую строку”

3)Вызов функции (-8, 8, 4)  
 a)Цикл -8 -4 0 4 8  
 -8: res1(-8, 2) = -8/2 = -4

Вывод: “tab”-8”tab”-4”переход на другую строку”

-4: res1(-4, 2) = -4/2 = -2

Вывод: “tab”-4”tab”-2”переход на другую строку”

0: Вывод: “tab”0”tab”0”переход на другую строку”

4: res2(4, 3) = 4/3 = 1.333333  
 Вывод: “tab”4”tab”1.333333”переход на другую строку”

8: res3(8, 3) = 8/4 = 2.666666

Вывод: “tab”8”tab”2.666666”переход на другую строку”  
 b)Возврат в main