

Примеры решения задач

1.1. Примеры работы с рекурсивными функциями

Пример 1.1. Создать рекурсивную функцию, которая для заданных целых A, B определяет функцию $NOD(A, B)$ по следующей формуле:

$$NOD(A, B) = \begin{cases} B & A = B \\ NOD(A - B, A) & A > B \\ NOD(A, B - A) & A < B \end{cases} \quad (1.1)$$

Листинг 1.1. Код программы

```

1  #include<iostream>
2  using namespace std;
3
4  int NOD(int a, int b){
5      if (a == b) return a; //результат
6      else if (a > b) {
7          cout << a << "\t" << b << endl; //вывод текущих данных
8          return NOD(a - b, b);
9      }
10     else {
11         cout << a << "\t" << b << endl; //вывод текущих данных
12         return NOD(a, b - a);
13     }
14 }
15
16 int main(){
17     int a,b;      //ввод чисел
18     cout << " a=";
19     cin >> a;
20     cout << " b=";
21     cin >> b;
22     cout << "a\tb\n"; //печать заголовка
23     cout << "NOD=" << NOD(a,b) << endl; //вызов функции
24     return 0;

```

Результат работы программы:

```

C:\WINDOWS\system32\cmd.exe
a=72
b=10
a      b
72     10
62     10
52     10
42     10
32     10
22     10
12     10
2       10
2       8
2       6
2       4
MOD=2
Для продолжения нажмите любую клавишу . . .
  
```

□

Пример 1.2. Создать рекурсивную функцию, которая для заданного N определяет значение функции

$$S = \sqrt{1 + \sqrt{2 + \sqrt{3 + \sqrt{4 + \cdots + \sqrt{n}}}}}$$

Эту функцию можно представить в следующем виде:

$$F(i, n) = \begin{cases} \sqrt{i + F(i+1, n)} & i < n \\ \sqrt{n} & i = n \end{cases}$$

Листинг 1.2. Код программы

```

1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4
5 double F(double i, double N){
6     if (abs(i - N) < 0.00001) return sqrt(N); //выход из рекурсии
7     else
8         return sqrt(i + F(i + 1, N));
9 }
  
```

```

10
11 int main(){
12     double N;
13     cout << " N = "; //Ввод данных
14     cin >> N;
15     cout << "F = " << F(1, N) << endl; //вызов функции
16     return 0;
17 }

```

Результат работы программы:

N	$F(N)$
2	1.55377
4	1.74876
8	1.75793

□

Пример 1.3. Вывести на экран следующую таблицу, используя рекурсивные функции:

```

111...111    (n раз)
22...22      (n - 2 раза)
3...3        (n - 4 раза)
...
n            (1 раз)
n            (1 раз)
...
3...3        (n - 4 раза)
22...22      (n - 2 раза)
111...111    (n раз)

```

Очевидно, что такая фигура может получиться только для однозначных чисел (можно напечатать максимум 9 строк, следовательно, n должно быть меньше 18).

Чтобы в центре получилась строка, содержащая одно число, n должно быть нечетным.

Видно, что первая строка содержит 0 пробелов и n символов, вторая — 1 пробел и $n - 2$ символа и т. д.

Условие выхода из рекурсии — $n = 0$.

Если функцию создать следующим образом:

1. Печать строки из p пробелов, n символов типа `char`.
2. Вызов функции $F(p + 1, n - 2, a + 1)$.

3. Если $n = 0$, переход к шагу 4, иначе переход к шагу 2.

4. Печать строки из p пробелов, n символов типа `char`.

Таким образом, сначала будет напечатана верхняя часть списка, потом нижняя.

Листинг 1.3. Код программы

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4
5 void print (int n, char a){ //печать n символов a
6     for (int i = 0; i < n; i++)
7         cout << a;
8 }
9
10 void F(int p, int n, char a){ // p - пробелов, n - символов
11     if (n > 0){
12         print(p, ' '); //печать пробелов
13         print (n, a); //печать цифр
14         cout << endl; //переход на следующую строку
15         F(p + 1, n - 2, a + 1); //вызов функции для следующей строки
16         print(p, ' '); //нижняя часть
17         print (n, a);
18         cout << endl;
19     }
20 }
21
22 int main(){
23     int N;
24     do{
25         cout << " Input N < 18 \n"; //ввод данных
26         cin >> N;
27     }
28     while (N > 18);
29     if (N%2 == 0) N --;
30     F(1, N, '1'); //вызов функции
31     return 0;
32 }
```

□

1.2. Шаблоны функций

Пример 1.4. В двумерном массиве поменять местами две средних строки, если количество строк четное, или первую и среднюю строку, если количество строк нечетное.

Программа будет содержать три функции: создание массива, замена строк и вывод массива на экран. В основной функции будет создан простейший интерфейс.

Листинг 1.4. Код программы

```
1 #include<iostream>
2 using namespace std;
3
4 template <typename X>
5 X **create(X c, int n, int m){//создание
6     X **a = new X *[n]; //выделение памяти
7     for (int i = 0; i < n; i++)
8         a[i] = new X [m];
9     for (int i = 0; i < n; i++) //ввод массива
10         for (int j = 0; j < m; j++)
11             cin >> a[i][j];
12     return a;
13 }
14
15 template <typename X>
16 void print(X **a, int n, int m){ //печать
17     for (int i = 0; i < n; i++,cout << endl)
18         for (int j = 0; j < m; j++)
19             cout << a[i][j] << " ";
20 }
21
22 template<typename X>
23 X ** change(X **a, int n, int m){//замена
24     if (n%2 == 0){ //четное
25         for (int j = 0; j < m; j++)
26             swap(a[n/2][j], a[n/2 - 1][j]);
27     }
28     else //нечетное
29         for (int j = 0; j < m; j++)
30             swap(a[n/2][j], a[0][j]);
31     return a;
```

```

32 }
33
34 int main(){
35     setlocale (LC_ALL, "rus");
36     int n,m;
37     cout << "Input dimension\n";
38     cin >> n >> m;
39     char c;
40     do{
41         cout << "Введите тип:\n1 - int\n2 - double\n3 - char\n4 - float\n";
42         int k;
43         cin >> k;
44         switch(k){//вызываем различные типы
45             case 1:
46                 {
47                     int c = 1;
48                     int **a = create(c, n, m);
49                     cout << "INT:\n";
50                     print(a, n, m);
51                     a = change(a, n, m);
52                     cout << "New:\n";
53                     print(a, n, m);
54                     break;
55                 }
56             case 2:
57                 {
58                     double c1 = 1;
59                     double **a1 = create(c1, n, m);
60                     cout << "DOUBLE:\n";
61                     print(a1, n, m);
62                     a1 = change(a1, n, m);
63                     cout << "New:\n";
64                     print(a1, n, m);
65                     break;
66                 }
67             case 3:
68                 {
69                     char c2 = '1';
70                     char **a2 = create(c2, n, m);

```

```

71     cout << "CHAR:\n";
72     print(a2, n, m);
73     a2 = change(a2, n, m);
74     cout << "New:\n";
75     print(a2, n, m);
76     break;
77 }
78 case 4:
79 {
80     float c3 = 1;
81     float **a3 = create(c3, n, m);
82     cout << "FLOAT:\n";
83     print(a3, n, m);
84     a3 = change(a3, n, m);
85     cout << "New:\n";
86     print(a3, n, m);
87     break;
88 }
89 default: cout << "Неправильно введены данные\n";
90 }
91 cout << "Еще раз? Y - да, N - нет\n";
92 cin >> c; // повторяем запуск программы пока не введем N
93 }
94 while(c != 'N');
95 return 0;
96 }

```

□