Обработка одномерного массива

# Зависимости.

## Для выполнения данной работы требуются следующие библиотеки:

1. #include <iostream>
2. #include <cstdlib>
3. #include <ctime>

# Задание 1.

1. **void** task\_1(){
2. /\*
3. Найти сумму всех элементов массива целых чисел,
4. удовлетворяющих условию: остаток от деления на 3 равен 2.
5. Размерность массива – 20.
6. \*/
8. srand (time(NULL));
9. **int** N = 20;
10. **int** a[N];
11. **for** ( **int** i = 1; i <= N; i++ ){ a[i] = rand(); }
13. **int** summ = 0;
14. **for** ( **int** i = 1; i <= N; i++ ){
15. **if** ( (a[i] % 3) == 2 ){ summ += a[i]; }
16. }
18. cout << summ << endl;
19. }

# Задание 2

1. **void** task\_2(){
2. /\*
3. Дан линейный массив целых чисел.
4. Проверьте, является ли он упорядоченным по убыванию.
5. \*/
7. srand (time(NULL));
8. **int** N = 3;; **int** a[N];
9. **for** ( **int** i = 1; i <= N; i++ ){ a[i] = rand(); }
11. cout << "- My massive otsortirovan po ubivanuy?" << endl;
12. **for** ( **int** i = 1; i <= N; i++ ){
13. **if**( i < N && a[i] < a[i+1] ){
14. cout << "- This is false..." << endl;
15. **break**;
16. }
17. **if** ( (i + 1) > N ){ cout << "- This is true" << endl; }
18. }
19. }

# Задание 3

1. **void** task\_3(){
2. /\*
3. Задан массив D. Определить следующие суммы:
4. D[l] + D[2] + D[3]; D[3] + D[4] + D[5]; D[4] + D[5] + D[6].
5. Пояснение. Составить подпрограмму для вычисления суммы трех
6. последовательно расположенных элементов массива с номерами от k до т.
7. \*/
9. srand (time(NULL));
10. **int** k,t; cout << "Input k and t: "; cin >> k >> t;
12. **int** N = 10; **int** D[N];
13. **for** ( **int** i = 1; i <= N; i++ ){ D[i] = rand() %10 + 1; }
15. **for** ( **int** i = k; i <= t; i++ ){
16. **if**( (i + 2) > t || (i + 1) >= t){ **break**; }
17. cout << ( D[i] + D[i+1] + D[i+2] ) << endl;
18. }
20. cout << "---------------" << endl;
21. **for** ( **int** i = 1; i <= N; i++ ){ cout << i << " | " << D[i] << endl; }
22. }

# Задание 4

1. **void** task\_4(){
2. /\*
3. Одномерный массив 12 элементов заполнить
4. случайными положительными и отрицательными значениями.
5. Переставить местами первый положительный элемент с последним отрицательным.
6. \*/
8. srand (time(NULL));
9. **int** N = 12; **int** D[N];
10. **for** ( **int** i = 1; i <= N; i++ ){ D[i] = rand() %1000 - 500; }
12. **int** sub1, sub2;
13. **for** ( **int** i = 1; i <=N; i++ ){
14. **if** ( D[i] >= 0){
15. sub1 = D[i];
17. **for** ( **int** ii = N; ii >= 1; i--){
18. **if** ( D[ii] < 0){
19. sub2 = D[ii];
20. D[ii] = sub1;
21. **break**;
22. }
23. }
25. D[i] = sub2;
26. **break**;
27. }
28. }
30. }

# Дополнительное задание 1

1. **void** add\_task\_1(){
2. /\*
3. Задана таблица названий товаров, выпускаемых заводом.
4. Определите, повторяется ли в этой таблице название первого товара,
5. и, если повторяется, удалите название первого товара из таблицы.
6. \*/
8. srand (time(NULL));
9. **int** N = 10; **int** D[N];
10. **for** ( **int** i = 1; i <= N; i++ ){ D[i] = rand()%1000 + 1000; }
12. **int** sub = D[1];
13. **for** ( **int** i = 2; i <= N; i++ ){
14. **if** ( sub == D[i] ){
15. **for** (**int** ii = 1; ii <= (N - 1); ii++ ){
16. **if** (ii == N){
17. D[ii] = 0;
18. }
19. **else**{
20. D[ii] = D[ii + 1];
21. }
22. }
23. }
24. **break**;
25. }
26. }

# Дополнительное задание 2

1. **void** add\_task\_2(){
2. /\*
3. Дан массив чисел. Найти, сколько в нем пар одинаковых соседних элементов.
4. \*/
5. srand (time(NULL));
6. **int** N = 10; **int** D[N];
7. **for** ( **int** i = 1; i <= N; i++ ){ D[i] = rand(); }
9. **int** num = 0;
10. **for** ( **int** i = 1; i <= N; i++ ){
11. **if** ( D[i] == D[i + 1] ){
12. num++;
13. }
14. }
16. cout << num << endl;;
17. }

# Сложное задание

1. **int** stalker(**int** N[20], **int** len, **int** b){
2. **for** ( **int** i = 1; i <= len; i++ ){
3. **if** ( N[i] == b){ **return** 0; }
4. }
5. **return** 1;
6. }
8. **void** hard\_task(){
9. /\*
10. Даны одномерные массивы А(состоит из 20 элементов) и В(из 10).
11. Сформировать массивы, состоящие из элементов:
12. 1) равных и в А, и в В
13. 2) А, которых нет в В
14. 3) В, которых нет в А
15. \*/
17. srand (time(NULL));
18. **int** D = 20, N = 10; **int** A[D], B[N];
19. **for** ( **int** i = 1; i <= D; i++ ){ A[i] = rand()%10 +1; }
20. **for** ( **int** i = 1; i <= N; i++ ){ B[i] = rand()%10 +1; }
22. // ==
23. **int** NA[D], k = 1;
24. **for** ( **int** i = 1; i <= D; i++ ){ NA[i] = 0; }
25. **for** ( **int** i = 1; i <= D; i++ ){
26. **for** ( **int** ii = 1; ii <= N; ii++ ){
27. **if** ( (A[i] == B[ii]) && stalker(NA, 20, A[i]) == 1 ){ NA[k] = A[i]; k++; **break**;}
28. }
29. }
30. //==
32. // Есть в А но нет в B
33. **int** NoB[D], kk = 1;
34. **for** ( **int** i = 1; i <= D; i++ ){ NoB[i] = 0; }
35. **for** ( **int** i = 1; i <= D; i++ ){
36. **if** ( stalker(NA, D, A[i]) == 1 ){ NoB[kk] = A[i]; kk++; }
37. }
38. // Есть в А но нет в B
40. **int** NoA[D], kkk = 1;
41. **for** ( **int** i = 1; i <= D; i++ ){ NoA[i] = 0; }
42. **for** ( **int** i = 1; i <= D; i++ ){
43. **if** ( stalker(NA, D, B[i]) == 1 ){ NoA[kkk] = A[i]; kkk++; }
45. }
46. cout << "In A and B: " << endl;
47. **for** ( **int** i = 1; i <= (k - 1); i++ ){ cout << i << " | " << NA[i] << endl; }
48. cout << "In A, but not in B: " << endl;
49. **for** ( **int** i = 1; i <= (kk -1); i++ ){ cout << i << " | " << NoB[i] << endl; }
50. cout << "IN B, but not in A: " << endl;
51. **for** ( **int** i = 1; i <= (kkk -1); i++ ){ cout << i << " | " << NoA[i] << endl; }
52. }