**Практическая работа №15**

**Вариант №2 – Расстановка скобок.**

**Автор: Николаев-Аксенов И. С.**

**Группа: ИКБО-20-19**

**Код программы:**

1. #include <iostream>
2. #include <limits.h>
3. using namespace std;
5. void printParenthesis(int i, int j, int n, int \*bracket, char &name)
6. {
7. if (i == j)
8. {
9. cout << name++;
10. return;
11. }
13. cout << "(";
15. printParenthesis(i, \*((bracket + j \* n) + i), n, bracket, name);
17. printParenthesis(\*((bracket + j \* n) + i) + 1, j, n, bracket, name);
18. cout << ")";
19. }
21. void matrixChainOrder(int p[], int n)
22. {
23. int min[n][n];
24. int max[n][n];
26. for (int i = 1; i < n; i++)
27. {
28. min[i][i] = 0;
29. max[i][i] = 0;
30. }
32. for (int L = 2; L < n; L++)
33. {
34. for (int i = 1; i < n - L + 1; i++)
35. {
36. int j = i + L - 1;
37. min[i][j] = INT\_MAX;
38. max[i][j] = INT\_MIN;
39. for (int k = i; k <= j - 1; k++)
40. {
41. int q = min[i][k] + min[k + 1][j] + p[i - 1] \* p[k] \* p[j];
42. if (q < min[i][j])
43. {
44. min[i][j] = q;
45. min[j][i] = k;
46. }
48. if (q >= max[i][j])
49. {
50. max[i][j] = q;
51. max[j][i] = k;
52. }
53. }
54. }
55. }
57. char matrixName = 'A';
58. cout << "Оптимальная расстановка скобок: ";
59. printParenthesis(1, n - 1, n, (int \*) min, matrixName);
60. cout << "**\n**Минимальное количество скалярных операций : " << min[1][n - 1] << endl;
62. matrixName = 'A';
63. cout << "**\n**Неоптимальная расстановка скобок: ";
64. printParenthesis(1, n - 1, n, (int \*) max, matrixName);
65. cout << "**\n**Максимальное количество скалярных операций : " << max[1][n - 1];
66. }

69. int main()
70. {
71. setlocale(LC\_ALL, "Russian");
73. int arr[] = {5, 10, 3, 12, 5, 50, 6};
75. int n = sizeof(arr) / sizeof(arr[0]);
76. matrixChainOrder(arr, n);
78. return 0;
79. }

**Результат выполнения программы:**

