

Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа фундаментальных физических исследований

Курсовой проект

Использование динамической памяти
По дисциплине "Специальный практикум"

Выполнил
студент гр. 5040302/10301

С. А. Буланова

Научный руководитель:
д.ф.-м.н.

Я. А. Бердников

«_____» _____ 2021 г.

Санкт-Петербург
2021

Задание

1. Создание динамического массива

Создать динамический массив типа `int` в стиле C. Написать функции вывода массива, сортировки, изменения размера массива и добавления нового элемента.

2. Класс для работы с динамическим массивом

Создать класс, содержащий в себе функции из предыдущего задания.

Задание 1

Приведем код:

```
1  #include <stdlib.h>
2  #include <stdint.h>
3  #include <stdio.h>
4
5  typedef int ValueType_t;
6
7  void PrintArray(const ValueType_t * p, const size_t nElements){
8      printf("Array: ");
9      for (size_t i = 0; i < nElements; i++){
10         printf("%5i", p[i]);
11     }
12     printf("\n");
13 }
14
15 void StupidSort(int *arr, int left, int right){
16     int i = left, j = right;
17     int cng, pivot = arr[(left+right)/2];
18     while (i <= j)
19     {
20         while (arr[i] < pivot) i++;
21         while (arr[j] > pivot) j--;
22
23         if (i <= j)
24         {
25             if (arr[i] > arr[j])
26             {
27                 cng = arr[i];
28                 arr[i] = arr[j];
29                 arr[j] = cng;
30             }
31             i++; j--;
32         }
33     }
34     if (left < j) StupidSort (arr, left, j);
35     if (i < right) StupidSort (arr, i, right);
36 }
37
```

```

38 void SortArray(ValueType_t * p, const size_t nElements){
39     StupidSort(p, 0, nElements);
40     printf("Sorted Array: ");
41     for (int i = 0; i < nElements; i++){
42         printf("%5i", p[i]);
43     }
44     printf("\n");
45 }
46
47 void ResizeArray(ValueType_t * p, size_t & nElements, const size_t nElementsNew){
48     printf("New Array: ");
49     for (size_t i = nElements; i < nElementsNew; i++){
50         p[i] = 0;
51     }
52
53     for (size_t i = 0; i < nElementsNew; i++){
54         printf("%5i", p[i]);
55     }
56     printf("\n");
57 }
58
59 void AppendArray(ValueType_t * p, size_t & nElements, ValueType_t Element){
60     printf("With Extra Elements: ");
61     for (size_t i = nElements; i < nElements + 1; i++){
62         p[i] = Element;
63     }
64
65     for (size_t i = 0; i < nElements + 1; i++){
66         printf("%5i", p[i]);
67     }
68     printf("\n");
69 }
70
71 int main(){
72     size_t arrsize;
73     size_t newsize;
74     size_t extrael;
75
76     ValueType_t * arr = new ValueType_t[100 * sizeof(int)];
77     for (size_t i = 0; i < 100 * sizeof(int); i++){

```

```

78     arr[i] = 0;
79 }
80 printf("Enter array size: ");
81 scanf("%zu", &arrsize);
82 for (int i = 0; i < arrsize; i++){
83     arr[i] = rand() %1000;
84 }
85
86 PrintArray(arr, arrsize);
87 printf("\n");
88 SortArray(arr, arrsize);
89 printf("\n");
90 newsize = arrsize * 1.5;
91 ResizeArray(arr, arrsize, newsize);
92 printf("\n");
93 printf("Enter extra element: ");
94 scanf("%zu", &extrael);
95 AppendArray(arr, arrsize, extrael);
96 delete [] arr;
97 }

```

Программа создает динамический массив, заполняет его случайными числами. Затем массив выводится, сортируется, изменяется его размер на какое-то число newsize (значения в промежутке arrsize и newsize заполняются нулями). Реализована функция добавления еще одного значения в конец массива.

Проверим, что ошибки компиляции отсутствуют, с помощью приложения valgrind.

```

==18842==
==18842== HEAP SUMMARY:
==18842==      in use at exit: 0 bytes in 0 blocks
==18842==    total heap usage: 4 allocs, 4 frees, 76,352 bytes allocated
==18842==
==18842== All heap blocks were freed -- no leaks are possible
==18842==
==18842== For lists of detected and suppressed errors, rerun with: -s
==18842== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
sonya@sonya-HP-Laptop-15-db0xxx:~/university/third_pair$

```

Посмотрим на вывод программы:

```
sonya@sonya-HP-Laptop-15-db0xxx:~/university/third_pair$ ./dyn
Enter array size: 55
Array:  383  886  777  915  793  335  386  492  649  421  362  27  690  59  763
       926  540  426  172  736  211  368  567  429  782  530  862  123  67  135  929
       802  22  58  69  167  393  456  11  42  229  373  421  919  784  537  198  32
       4  315  370  413  526  91  980  956

Sorted Array:  0  11  22  27  42  58  59  67  69  91  123  135  167  1
              72  198  211  229  315  324  335  362  368  370  373  383  386  393  413  421  421
              426  429  456  492  526  530  537  540  567  649  690  736  763  777  782  784
              793  802  862  886  915  919  926  929  956

New Array:  0  11  22  27  42  58  59  67  69  91  123  135  167  172
            198  211  229  315  324  335  362  368  370  373  383  386  393  413  421  421  4
            26  429  456  492  526  530  537  540  567  649  690  736  763  777  782  784  793
            802  862  886  915  919  926  929  956  0  0  0  0  0  0  0  0
            0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
            0  0  0

Enter extra element: 9999
With Extra Elements:  0  11  22  27  42  58  59  67  69  91  123  135
                    167  172  198  211  229  315  324  335  362  368  370  373  383  386  393  413  4
                    21  421  426  429  456  492  526  530  537  540  567  649  690  736  763  777  782
                    784  793  802  862  886  915  919  926  929  956  9999
sonya@sonya-HP-Laptop-15-db0xxx:~/university/third_pair$
```

Задание 2

Реализуем похожую программу, но с использованием классов. Код:

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4
5  typedef int ValueType_t;
6
7  class DynamicArray{
8  public:
9
10     void PrintArray(const ValueType_t * p, const size_t nElements){
11         cout << "Array: ";
12         for (size_t i = 0; i < nElements; i++){
13             cout << p[i] << ", ";
14         }
15         cout << '\n';
16     }
17
18     void StupidSort(int *arr, int left, int right){
19         int i = left, j = right;
```

```

20     int cng, pivot = arr[(left+right)/2];
21     while (i <= j)
22     {
23         while (arr[i] < pivot) i++;
24         while (arr[j] > pivot) j--;
25
26         if (i <= j)
27         {
28             if (arr[i] > arr[j])
29             {
30                 cng = arr[i];
31                 arr[i] = arr[j];
32                 arr[j] = cng;
33             }
34             i++; j--;
35         }
36     }
37     if (left < j) StupidSort (arr, left, j);
38     if (i < right) StupidSort (arr, i, right);
39 }
40
41 void SortArray(ValueType_t * p, const size_t nElements){
42     StupidSort(p, 0, nElements);
43     cout << "Sorted Array: ";
44     for (int i = 0; i < nElements; i++){
45         cout << p[i] << ", ";
46     }
47     cout << '\n';
48 }
49
50 void ResizeArray(ValueType_t * p, size_t & nElements, const size_t nElementsNew){
51     cout << "New Array: ";
52     for (size_t i = nElements; i < nElementsNew; i++){
53         p[i] = 0;
54     }
55     for (size_t i = 0; i < nElementsNew; i++){
56         cout << p[i] << ", ";
57     }
58     cout << '\n';
59 }

```

```

60
61 void AppendArray(ValueType_t * p, size_t & nElements, ValueType_t Element){
62     cout << "With Extra Elements: ";
63     for (size_t i = nElements; i < nElements + 1; i++){
64         p[i] = Element;
65     }
66     for (size_t i = 0; i < nElements + 1; i++){
67         cout << p[i] << ", ";
68     }
69     cout << '\n';
70 }
71
72 ~DynamicArray() {}
73 };
74
75 int main(){
76     size_t nElements;
77     size_t newsize;
78     size_t extrael;
79
80     ValueType_t * p = new ValueType_t[100 * sizeof(int)];
81     for (size_t i = 0; i < 100 * sizeof(int); i++){
82         p[i] = 0;
83     }
84
85     cout << "Enter array size: ";
86     cin >> nElements;
87     for (int i = 0; i < nElements; i++){
88         p[i] = rand() %10000;
89     }
90     DynamicArray printarray;
91
92     printarray.PrintArray(p, nElements);
93     cout << '\n';
94     printarray.SortArray(p, nElements);
95     cout << '\n';
96     newsize = nElements * 1.5;
97     printarray.ResizeArray(p, nElements, newsize);
98     cout << '\n';
99     cout << "Enter extra element: ";

```

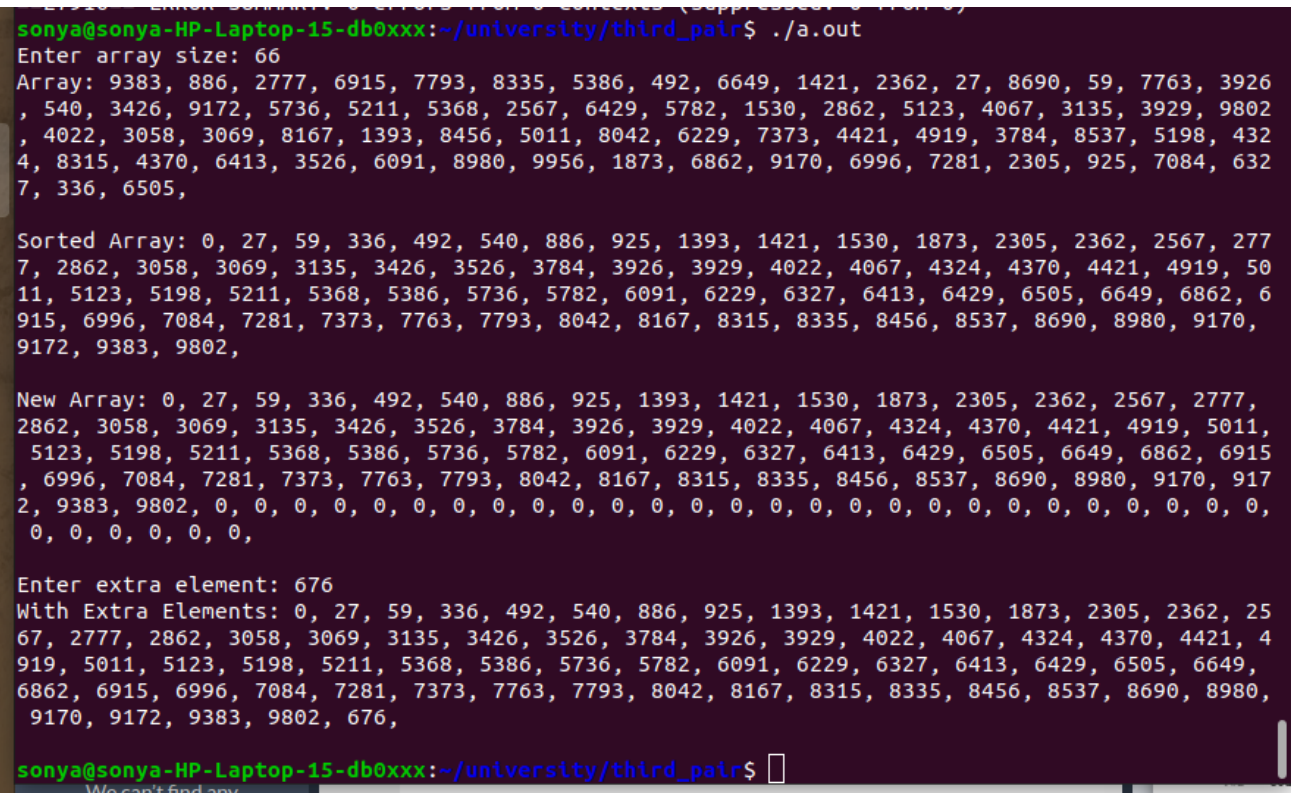


```

100     cin >> extrael;
101     printarray.AppendArray(p, nElements, extrael);
102     cout << '\n';
103
104     delete [] p;
105 }

```

Результат работы:



```

sonya@sonya-HP-Laptop-15-db0xxx:~/university/third_pair$ ./a.out
Enter array size: 66
Array: 9383, 886, 2777, 6915, 7793, 8335, 5386, 492, 6649, 1421, 2362, 27, 8690, 59, 7763, 3926
, 540, 3426, 9172, 5736, 5211, 5368, 2567, 6429, 5782, 1530, 2862, 5123, 4067, 3135, 3929, 9802
, 4022, 3058, 3069, 8167, 1393, 8456, 5011, 8042, 6229, 7373, 4421, 4919, 3784, 8537, 5198, 432
4, 8315, 4370, 6413, 3526, 6091, 8980, 9956, 1873, 6862, 9170, 6996, 7281, 2305, 925, 7084, 632
7, 336, 6505,

Sorted Array: 0, 27, 59, 336, 492, 540, 886, 925, 1393, 1421, 1530, 1873, 2305, 2362, 2567, 277
7, 2862, 3058, 3069, 3135, 3426, 3526, 3784, 3926, 3929, 4022, 4067, 4324, 4370, 4421, 4919, 50
11, 5123, 5198, 5211, 5368, 5386, 5736, 5782, 6091, 6229, 6327, 6413, 6429, 6505, 6649, 6862, 6
915, 6996, 7084, 7281, 7373, 7763, 7793, 8042, 8167, 8315, 8335, 8456, 8537, 8690, 8980, 9170,
9172, 9383, 9802,

New Array: 0, 27, 59, 336, 492, 540, 886, 925, 1393, 1421, 1530, 1873, 2305, 2362, 2567, 2777,
2862, 3058, 3069, 3135, 3426, 3526, 3784, 3926, 3929, 4022, 4067, 4324, 4370, 4421, 4919, 5011,
5123, 5198, 5211, 5368, 5386, 5736, 5782, 6091, 6229, 6327, 6413, 6429, 6505, 6649, 6862, 6915
, 6996, 7084, 7281, 7373, 7763, 7793, 8042, 8167, 8315, 8335, 8456, 8537, 8690, 8980, 9170, 917
2, 9383, 9802, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0,

Enter extra element: 676
With Extra Elements: 0, 27, 59, 336, 492, 540, 886, 925, 1393, 1421, 1530, 1873, 2305, 2362, 25
67, 2777, 2862, 3058, 3069, 3135, 3426, 3526, 3784, 3926, 3929, 4022, 4067, 4324, 4370, 4421, 4
919, 5011, 5123, 5198, 5211, 5368, 5386, 5736, 5782, 6091, 6229, 6327, 6413, 6429, 6505, 6649,
6862, 6915, 6996, 7084, 7281, 7373, 7763, 7793, 8042, 8167, 8315, 8335, 8456, 8537, 8690, 8980,
9170, 9172, 9383, 9802, 676,

sonya@sonya-HP-Laptop-15-db0xxx:~/university/third_pair$

```