

Họ tên: Nguyễn Phạm Thành Hưng

MSSV: 6151071056

Ngày sinh: 16-09-2002

Email: 6151071056@st.utc2.edu.vn

BÀI TẬP THỰC HÀNH CTDL

CÁC THUẬT TOÁN SẮP XẾP VÀ TÌM KIẾM

Yêu cầu:

1. Sinh viên điền đầy đủ thông tin cá nhân ở đầu trang.
2. Sinh viên viết chương trình C++ hoàn chỉnh cho các bài tập bên dưới.
3. Source code mỗi câu phải chụp hình dán vào word (Insert → Screenshot → ScreenClipping) và đầu mỗi phần code phải có đủ thông tin họ tên, MSSV, ngày sinh và email (xem ví dụ ở hình sau).

```
// Ho ten: Nguyen Minh Ngoc
// MSSV: 123456
// Ngay sinh: 1/1/2000
// Email: ngoc@gmail.com
// Ham xuất mảng bang de quy
void XuatMang(int a[], int n)
{
    // Diem dung
    if (n == 0)
        return;
    // Goi de quy
    XuatMang(a, n-1);
    printf("%d\t", a[n-1]);
}
```

4. Sau khi hoàn thành, sinh viên chuyển file word sang file PDF và sử dụng file PDF để nộp bài.
Đặt tên file theo quy tắc sau: <4 số cuối của mã sinh viên>_<Họ tên không dấu>.pdf
Ví dụ: 3456_NguyenMinhNgoc.pdf
5. Thời hạn nộp bài: 11.59 Thứ 6 (26/11/2021).

Chú ý: Sinh viên bị trừ điểm trong các trường hợp sau:

- Thiếu thông tin cá nhân tại source code (-0.5đ cho mỗi lần thiếu).
- Không nộp file PDF (-1đ).
- Tên file không theo đúng quy định (-1đ).
- Nộp bài trễ (cứ mỗi 30 phút, trừ 1đ).

LƯU Ý: TẤT CẢ CÁC BÀI PHẢI VIẾT DƯỚI DẠNG HÀM

Bài 1. Danh sách sử dụng mảng tĩnh: int A[1000];

Cho một dãy số nguyên. Giả sử dãy số đó được lưu tại một mảng tĩnh.

1. Hãy cài đặt các thuật toán sắp xếp đã học.

```
20 //Ho va Ten: Nguyen Pham Thanh Hung
21 //MSSV: 6151071056
22 //Ngày sinh: 16/09/2002
23 //Email: 6151071056@st.utc2.edu.vn
24 void BubbleSort(int A[], int n){
25     for(int i=0; i<n; i++){
26         for(int j=0; j<n-i-1; j++){
27             if(A[j]>A[j+1]){
28                 swap (A[j], A[j+1]);
29             }
30         }
31     }
32 }
33 //Do phuc tap Bubble Sort O(n2)
34
35 void insertionSort(int A[], int n){
36     for(int i=1; i<n; i++){
37         int x = A[i];
38         int j = i-1;
39         while (j>=0 && A[j]>x){
40             A[j+1] = A[j];
41             j--;
42         }
43         A[j+1] = x;
44     }
45 }
46 //Do phuc tap Insert Sort O(n2)
47
48 void selectionSort(int A[], int n){
49     for(int i=0; i<n-1; i++){
50         int minV = A[i], pMin = i;
51         for(int j = i+1; j<n; j++){
52             if(minV > A[j]){
53                 minV = A[j];
54                 pMin = j;
55             }
56             if(i!=pMin){
57                 swap(A[i], A[pMin]);
58             }
59         }
60     }
61 }
```

```

64 void QuickSort(int A[], int left, int right){
65     int x = A[(left + right)/2];
66     int i = left, j=right;
67     do{
68         while (A[i]<x) i++;
69         while (A[j]>x) j--;
70         if(i<=j){
71             swap(A[i], A[j]);
72             i++;
73             j--;
74         }
75     } while (i<j);
76
77     if(left<j){
78         QuickSort(A, left, j);
79     }
80
81     if(right>i){
82         QuickSort(A, i, right);
83     }
84 }

```

2. Hãy cài đặt các thuật toán tìm kiếm đã học.

```

87 //Ho va Ten: Nguyen Pham Thanh Hung
88 //MSSV: 6151071056
89 //Ngày sinh: 16/09/2002
90 //Email: 6151071056@st.utc2.edu.vn
91
92 int binarySearch(int A[], int n, int key){
93     int left = 0, right = n-1, mid;
94     while(left<=right){
95         mid = (left+right)/2;
96         if(A[mid]==key)
97             return mid;
98         if(A[mid]<key)
99             left = mid +1;
100         else
101             right = mid -1;
102     }
103     return -1;
104 }
105 //Độ phức tạp về thời gian: O (logn)
106 //Độ phức tạp không gian: O (1)
107
108 int linearSearch(int A[], int n, int key){
109     for(int i=0; i<n; i++){
110         if(A[i]==key)
111             return i;
112     }
113     return -1;
114 }

```

3. Viết hàm main để kiểm tra kết quả thực hiện của các hàm trên.

```

118 //Ho va Ten: Nguyen Pham Thanh Hung
119 //MSSV: 6151071056
120 //Ngày sinh: 16/09/2002
121 //Email: 6151071056@st.utc2.edu.vn
122
123 int main(){
124     int A[1000], n;
125     int left, right;
126     int key;
127     cout << "Nhap vao so luong phan tu: ";
128     cin >> n;
129     for(int i=0; i<n; i++){
130         cout<< "Nhap phan tu thu " << i+1 << " : ";
131         cin >> A[i];
132     }
133     cout<< "Mang da nhap la: ";
134     printArray(A, n);
135     cout<< endl;
136     cout<< "Bubble Sort: ";
137     BubbleSort(A, n);
138     printArray(A,n);
139     cout<< endl;
140     cout << "Insert Sort: ";
141     insertionSort(A, n);
142     printArray(A, n);
143     cout<< endl;
144     cout<< "Selection Sort: ";
145     selectionSort(A, n);
146     printArray(A, n);
147     cout << endl;
148     cout << "Quick Sort: ";
149     QuickSort(A, left, right);
150     printArray(A, n);
151     cout << endl;
152     cout << "Nhap gia tri can tim kiem: ";
153     cin >> key;
154     int v = binarySearch(A, n, key);
155     cout << "Binary Search: " << v << endl;
156     int t = linearSearch(A, n, key);
157     cout << "Linear Search: " << t << endl;
158     system("pause");
159     return 0;
160 }

```

4. Hãy cho biết độ phức tạp của các thuật toán sắp xếp và sắp xếp các thuật toán đó theo chiều tăng dần của độ phức tạp.
 - Thuật toán sắp xếp nhanh là $O(n\log(n))$
 - Thuật toán sắp xếp chọn là $O(n^2)$
 - Thuật toán sắp xếp chèn là $O(n^2)$
 - Thuật toán sắp xếp nổi bọt là $O(n^2)$
5. Hãy cho biết độ phức tạp của các thuật toán tìm kiếm và sắp xếp các thuật toán đó theo chiều tăng dần của độ phức tạp.

```
Họ và Tên: Nguyen Pham Thanh Hung  
MSSV: 6151071056  
Ngày sinh: 16/09/2002  
Email: 6151071056@st.utc2.edu.vn
```

```
====binarySearch====
```

```
Độ phức tạp về thời gian: 0 (logn)
```

```
Độ phức tạp không gian: 0 (1)
```

```
=====linearSearch=====
```

```
Độ phức tạp về thời gian: 0 (n)
```

```
Độ phức tạp không gian: 0 (1)
```

Bài 2. Danh sách sử dụng mảng động

int *A;

Cấp phát bộ nhớ và Giải phóng bộ nhớ

Cho một dãy số nguyên. Giả sử dãy số đó được lưu tại một mảng động.

1. Hãy cài đặt các thuật toán sắp xếp đã học.

```

25 //Ho va Ten: Nguyen Pham Thanh Hung
26 //MSSV: 6151071056
27 //Ngày sinh: 16/09/2002
28 //Email: 6151071056@st.utc2.edu.vn
29 void BubbleSort(int *A, int n){
30     for(int i=0; i<n; i++){
31         for(int j=0; j<n-i-1; j++){
32             if(*(A+i)>*(A+j+1)){
33                 swap (*(A+i), *(A+j+1));
34             }
35         }
36     }
37 }
38 void insertSort(int *A, int n){
39     for(int i=1; i<n; i++){
40         int x = *(A+i);
41         int j = i-1;
42         while (j>=0 && *(A+j)>x){
43             *(A+j+1) = *(A+j);
44             j--;
45         }
46         *(A+j+1) = x;
47     }
48 }
49
50 void selectionSort(int *A, int n){
51     for(int i=0; i<n-1; i++){
52         int minV = *(A+i), pMin = i;
53         for(int j = i+1; j<n; j++){
54             if(minV > *(A+j)){
55                 minV = *(A+j);
56                 pMin = j;
57             }
58             if(i!=pMin){
59                 swap(*(A+i), *(A+pMin));
60             }
61         }
62     }
63 }

```

```

65 void QuickSort(int *A, int left, int right){
66     int x = *(A+left + right)/2;
67     int i = left, j=right;
68     do{
69         while (*(A+i)<x) i++;
70         while (*(A+j)>x) j--;
71         if(i<=j){
72             swap(*(A+i), *(A+j));
73             i++;
74             j--;
75         }
76     } while (i<j);
77
78     if(left<j){
79         QuickSort(A, left, j);
80     }
81
82     if(right>i){
83         QuickSort(A, i, right);
84     }
85 }

```

2. Hãy cài đặt các thuật toán tìm kiếm đã học.

```

87 //Ho va Ten: Nguyen Pham Thanh Hung
88 //MSSV: 6151071056
89 //Ngày sinh: 16/09/2002
90 //Email: 6151071056@st.utc2.edu.vn
91
92 int binarySearch(int *A, int n, int key){
93     int left = 0, right = n-1, mid;
94     while(left<=right){
95         mid = (left+right)/2;
96         if(*(A+mid)==key)
97             return mid;
98         if(*(A+mid)<key)
99             left = mid +1;
100         else
101             right = mid -1;
102     }
103     return -1;
104 }
105
106 int linearSearch(int *A, int n, int key){
107     for(int i=0; i<n; i++){
108         if(*(A+i)==key)
109             return i;
110     }
111     return -1;
112 }

```

3. Viết hàm main để kiểm tra kết quả thực hiện của các hàm trên.

```

114 //Ho va Ten: Nguyen Pham Thanh Hung
115 //MSSV: 6151071056
116 //Ngày sinh: 16/09/2002
117 //Email: 6151071056@st.utc2.edu.vn
118
119 int main()
120 {
121     cout << "Nhap so luong phan tu: ";
122     int n;
123     cin >> n;
124     int *A = new int[n];
125     int left, right;
126     int key;
127     inputArray(A, n);
128     cout<< "Mang da nhap la: ";
129     printArray(A, n);
130     cout<< endl;
131     cout<< "Bubble Sort: ";
132     BubbleSort(A, n);
133     printArray(A,n);
134     cout<< endl;
135     cout << "Insert Sort: ";
136     insertSort(A, n);
137     printArray(A, n);
138     cout<< endl;
139     cout<< "Selection Sort: ";
140     selectionSort(A, n);
141     printArray(A, n);
142     cout << endl;
143     cout << "Quick Sort: ";
144     QuickSort(A, left, right);
145     printArray(A, n);
146     cout << endl;
147     cout << "Nhap gia tri can tim kiem: ";
148     cin >> key;
149     int v = binarySearch(A, n, key);
150     cout << "Binary Search: " << v << endl;
151     int t = linearSearch(A, n, key);
152     cout << "Linear Search: " << t << endl;
153     delete[] A;
154
155     return 0;
156 }

```

Bài 3. Danh sách sử dụng danh sách liên kết đơn

Cho một dãy số nguyên. Giả sử dãy số đó được lưu tại một danh sách liên kết đơn.

1. Trong số các thuật toán sắp xếp và tìm kiếm đã học, các thuật toán nào có thể cài đặt được với danh sách liên kết đơn.
 - Sắp xếp: Nổi bọt, chọn
 - Tìm kiếm: Tuyến tính
2. Hãy cài đặt các thuật toán tìm được ở câu 1.


```

87 //Ho va Ten: Nguyen Pham Thanh Hung
88 //MSSV: 6151071056
89 //Ngày sinh: 16/09/2002
90 //Email: 6151071056@st.utc2.edu.vn
91 //Bubble Sort
92 void BubbleSort(Slist sl) {
93     Node *p, *q;
94     p = sl.head;
95     while(p != NULL) {
96         q = p->next;
97         while(q != NULL) {
98             if(p->data > q->data)
99                 Swap(p->data, q->data);
100             q = q->next;
101         }
102         p = p->next;
103     }
104 }
105
106 //Selection Sort
107 void SelectionSort(Slist sl) {
108     Node *p, *q, *min;
109     p = sl.head;
110     while(p != NULL) {
111         min = p;
112         q = p->next;
113         while(q != NULL) {
114             if(min->data > q->data)
115                 min = q;
116             q = q->next;
117         }
118         Swap(p->data, min->data);
119         p = p->next;
120     }
121 }
122
123
124 //Ho va Ten: Nguyen Pham Thanh Hung
125 //MSSV: 6151071056
126 //Ngày sinh: 16/09/2002
127 //Email: 6151071056@st.utc2.edu.vn
128 //Linear Search
129 Item LinearSeach(Slist sl, Item key) {
130     Node* p = sl.head;
131     Item count = 0;
132     while(p != NULL) {
133         if(p->data == key)
134             return count;
135         count++;
136         p = p->next;
137     }
138     return -1;
139 }

```

3. Viết hàm main để kiểm tra kết quả thực hiện của các hàm trên.

```

141 //Ho va Ten: Nguyen Pham Thanh Hung
142 //MSSV: 6151071056
143 //Ngày sinh: 16/09/2002
144 //Email: 6151071056@st.utc2.edu.vn
145
146 int main() {
147     Slist list;
148     Item key;
149     if(list.head == NULL)
150         cout << "HEAD = NULL" << endl;
151     if(list.tail == NULL)
152         cout << "TAIL = NULL" << endl;
153     cout << endl;
154     list.addFirst(55);
155     list.addFirst(25);
156     list.addFirst(40);
157     list.addLast(45);
158     list.addLast(10);
159     list.Traverser();
160     cout << "\nBubble sort: " << endl;
161     BubbleSort(list);
162     list.Traverser();
163     cout << "\nSelection sort: " << endl;
164     SelectionSort(list);
165     list.Traverser();
166     cout << "\nNhap phan tu: "; cin >> key;
167     if(LinearSeach(list, key) == -1)
168         cout << "\nKhong co phan tu " << key << " trong danh sach!";
169     else
170         cout << "\nPhan tu " << key << " o vi tri: " << LinearSeach(list, key) << endl;
171 }

```

Bài 4.

Cho N điểm trên mặt phẳng tọa độ Oxy. Giả sử danh sách này được lưu trong một mảng (tĩnh hoặc động), mỗi phần tử có kiểu dữ liệu Point gồm 2 thành phần x và y.

```

struct Point {
    float x;
    float y;
}

```

1. Sắp xếp N điểm đó theo chiều tăng dần của khoảng cách từ điểm đó tới gốc tọa độ.

```

33 // Ho va Ten: Nguyen Pham Thanh Hung
34 // MSSV: 6151071056
35 // Ngay sinh: 16/09/2002
36 // Email: 6151071056@st.utc2.edu.vn
37 void sort(ToDo *a, int n){
38     for(int i=0;i<n-1;i++){
39         for(int j=0;j<n-i-1;j++){
40             if(length((a+j)) > length((a+j+1))){
41                 swap(*(a+j),*(a+j+1));
42             }
43         }
44     }
45 }

```

2. Hãy tìm các điểm nằm trong hình tròn tâm O bán kính R (R do người dùng nhập từ bàn phím).

```

47 // Ho va Ten: Nguyen Pham Thanh Hung
48 // MSSV: 6151071056
49 // Ngay sinh: 16/09/2002
50 // Email: 6151071056@st.utc2.edu.vn
51
52 void find(ToDo *a, int n, int R){
53     cout<<"\nCac diem nam trong duong tron tam O ban kinh R: "<<endl;
54     for(int i=0;i<n;i++){
55         if(length((a+i))<R) print(a+i);
56     }
57 }

```

3. Hãy tìm các cặp điểm thỏa mãn điều kiện hai điểm nằm trên hai phía của trục hoành.

```

59 // Ho va Ten: Nguyen Pham Thanh Hung
60 // MSSV: 6151071056
61 // Ngay sinh: 16/09/2002
62 // Email: 6151071056@st.utc2.edu.vn
63
64 void findX(ToDo *a, int n){
65     for(int i=0;i<n;i++){
66         if(((a+i)->x>0 && (a+i)->y>0) || ((a+i)->x<0 && (a+i)->y>0)) print(a+i);
67     }
68 }

```

4. Hãy tìm các điểm có khoảng cách tới gốc tọa độ bằng d (d do người dùng nhập từ bàn phím).

```

70 // Ho va Ten: Nguyen Pham Thanh Hung
71 // MSSV: 6151071056
72 // Ngay sinh: 16/09/2002
73 // Email: 6151071056@st.utc2.edu.vn
74
75 void findD(ToDo *a, int n, int d){
76     int dem=0;
77     for(int i=0;i<n;i++){
78         if(length(a+i) == d) {
79             print(a+i);
80             dem++;
81         }
82     }
83     if(dem==0) cout<<"\nKhong co diem nao bang d";
84 }

```

5. Viết hàm main để kiểm tra kết quả thực hiện của các hàm trên.

```
86 // Ho va Ten: Nguyen Pham Thanh Hung
87 // MSSV: 6151071056
88 // Ngay sinh: 16/09/2002
89 // Email: 6151071056@st.utc2.edu.vn
90
91 int main(){
92     int n;
93     ToaDo *a=new ToaDo[n];
94     enterList(a,n);
95     sort(a,n);
96     float R;
97     cout<<"\nNhap ban kinh duong tron tam O: ";
98     cin>>R;
99     find(a,n,R);
100    cout<<"\nCac diem nam 2 phia ben tren truc hoành: ";
101    findX(a,n);
102    float d;
103    cout<<"\nNhap d: ";
104    cin>>d;
105    cout<<"\nCac diem co khoanh cach bang d: ";
106    findD(a,n,d);
107    delete[] a;
108    return 0;
109 }
```

Bài 5.

Cho danh sách gồm N cuốn sách. Mỗi cuốn sách có tên sách (xâu ký tự) và số trang (int).

1. Hãy sắp xếp các cuốn sách theo chiều tăng dần của tên sách.

```
35 // Ho va Ten: Nguyen Pham Thanh Hung
36 // MSSV: 6151071056
37 // Ngay sinh: 16/09/2002
38 // Email: 6151071056@st.utc2.edu.vn
39 //quick sort
40 void SortBook(Book *b, int left, int right){
41     Book x = *(b+((left + right)/2));
42     int i = left, j = right;
43     do{
44         while((b+i)->name < x.name)
45             i++;
46         while ((b+j)->name > x.name)
47             j--;
48         if(i <= j){
49             swap(*(b+i),*(b+j));
50             i++;
51             j--;
52         }
53     }while(i < j);
54     if(left < j)
55         SortBook(b,left,j);
56     if(right > i)
57         SortBook(b,i,right);
58 }
```

2. Hãy sắp xếp các cuốn sách theo chiều giảm dần của số trang.

```

60 // Ho va Ten: Nguyen Pham Thanh Hung
61 // MSSV: 6151071056
62 // Ngay sinh: 16/09/2002
63 // Email: 6151071056@st.utc2.edu.vn
64
65 //quick sort
66 void SortPage(Book *b, int left, int right){
67     Book x = *(b+((left + right)/2));
68     int i = left, j = right;
69     do{
70         while((b+i)->page > x.page)
71             i++;
72         while ((b+j)->page < x.page)
73             j--;
74         if(i <= j){
75             swap(*(b+i),*(b+j));
76             i++;
77             j--;
78         }
79     }while(i < j);
80     if(left < j)
81         SortPage(b, left, j);
82     if(right > i)
83         SortPage(b, i, right);
84 }

```

3. Tìm cuốn sách có tên X (X do người dùng nhập từ bàn phím).

```

86 // Ho va Ten: Nguyen Pham Thanh Hung
87 // MSSV: 6151071056
88 // Ngay sinh: 16/09/2002
89 // Email: 6151071056@st.utc2.edu.vn
90
91 //binary search
92 Book findBook(Book *b, int n, string name){
93     int left = 0, right = n-1, mid;
94     while (left <= right){
95         mid = (left + right)/ 2;
96         if ((b+mid)->name == name)
97             return b[mid];
98         if ((b+mid)->name < name)
99             left = mid+1;
100         else
101             right = mid-1;
102     }
103     Book nothing;
104     return nothing;
105 }

```

4. Liệt kê các cuốn sách có số trang ít hơn p (p do người dùng nhập vào từ bàn phím).

```

107 // Ho va Ten: Nguyen Pham Thanh Hung
108 // MSSV: 6151071056
109 // Ngay sinh: 16/09/2002
110 // Email: 6151071056@st.utc2.edu.vn
111
112 void statistics(Book *b, int n, int p){
113     int count = 0;
114     for(int i=0; i < n; i++){
115         if((b+i)->page < p){
116             output(b+i);
117             count++;
118         }
119     }
120     if(count == 0){
121         cout<<"Not found"<<endl;
122     }
123 }

```

5. Viết hàm main để kiểm tra kết quả thực hiện của các hàm trên.

```

125 // Ho va Ten: Nguyen Pham Thanh Hung
126 // MSSV: 6151071056
127 // Ngay sinh: 16/09/2002
128 // Email: 6151071056@st.utc2.edu.vn
129
130 int main(){
131     Book *b;
132     int n;
133     cout<<"Enter number of book: ";
134     cin>>n;
135     b = new Book[n];
136     cout <<endl<<"Book infomation"<<endl;
137     inputList(b, n);
138     SortBook(b, 0, n-1);
139     cout <<endl<<"Sort by name"<<endl;
140     for (int i=0; i < n; i++){
141         cout <<"Number: "<<i+1;
142         output(b+i);
143     }
144     SortPage(b, 0, n-1);
145     cout <<endl<<"Sort by page"<<endl;
146     for (int i=0; i < n; i++){
147         cout <<"Number: "<<i+1;
148         output(b+i);
149     }
150     string name;
151     cout<<"Enter book name to find: ";
152     fflush(stdin);
153     getline(cin, name);
154     if(findBook(b, n, name).name == " "){
155         cout<<"Not Found"<<endl;
156     }else output(b);
157     int p;
158     cout<<"Enter page need to statistic: ";
159     cin>>p;
160     statistics(b, n, p);
161     delete [] b;
162     return 0;
163 }

```

