Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων

Μάθημα: Δομές Δεδομένων και Αλγόριθμοι (Εργαστήριο)
Ακαδημαϊκό έτος 2018-2019

ΠΡΟΟΔΟΣ - ΔΕΚΕΜΒΡΙΟΣ 2018
Διάρκεια εξέτασης: 60 λεπτά

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Α.Μ.:

ΘΕΜΑΤΑ Α

Θέμα 1 [4 μονάδες]

1. Να συμπληρώσετε τον αλγόριθμο δυαδικής αναζήτησης στα σημεία Α, Β, Γ και Δ. Η συνάρτηση binary_search θα πρέπει να επιστρέφει τη θέση στην οποία βρίσκεται το στοιχείο key στον πίνακα a, λαμβάνοντας υπόψη τα στοιχεία του a από τη θέση left μέχρι και τη θέση right. Αν το στοιχείο key δεν υπάρχει, η συνάρτηση θα πρέπει να επιστρέφει την τιμή -1.

```
int binary_search(int a[], int left, int right, int key) {
 int m = (left + right) / 2;
 if (left > right) {
   [A] return -1;
 } else if (key < a[m]) {
   [B] return binary_search(a, left, m-1, key);
 } else if (key > a[m]) {
   [Γ] return binary_search(a, m+1, right, key);
 }
 else {
   [Δ]  return m;
 }
}
```

2. Έστω πίνακας 16 θέσεων που περιέχει τις τιμές {1,2,3,5,7,8,10,12,14,15,16,17,18,20,21,23}. Χρησιμοποιώντας τη δυαδική αναζήτηση, ποια θα είναι τα στοιχεία του πίνακα με τα οποία θα συγκριθεί η τιμή 19 έτσι ώστε ο αλγόριθμος να επιστρέψει -1, δηλώνοντας ότι το συγκεκριμένο κλειδί δεν υπάρχει στον πίνακα;
   Θα συγκριθεί με τις τιμές: 12, 17,20,18

3. Γράψτε τον κώδικα που δηλώνει και αρχικοποιεί τον πίνακα του προηγούμενου ερωτήματος, που πραγματοποιεί την κλήση της συνάρτησης για την αναζήτηση της τιμής 19 στον πίνακα και που εμφανίζει το αποτέλεσμα που επιστρέφει η συνάρτηση.

```
int a[]={1,2,3,5,7,8,10,12,14,15,16,17,18,20,21,23};
cout << binary_search(a, 0, 15, 19) << endl;
```

Θέμα 2 [3 μονάδες]

1. Για μια απλά συνδεδεμένη λίστα συμπληρώστε τον κώδικα [A] που περιγράφει έναν κόμβο της (struct forologumenos), ο οποίος αφορά ένα φορολογούμενο περιέχοντας τα στοιχεία: όνομα (onoma, τύπου string), αριθμός φορολογικού μητρώου (afm, τύπου string) και φορολογητέο εισόδημα (eisodima, τύπου int).
2. Συμπληρώστε τον κώδικα [B] που θα εμφανίζει τα ονόματα και τα ΑΦΜ για όλους τους φορολογούμενους με κάτω από 10000 € φορολογητέο εισόδημα.
3. Σε μια απλά συνδεδεμένη λίστα είναι ταχύτερο να εισάγουμε νέα στοιχεία στην αρχή, ή στο τέλος, ή δεν έχει διαφορά; Αιτιολογήστε την απάντησή σας.

   Είναι ταχύτερο να εισαχθεί στην αρχή καθώς δεν χρειάζεται να διανυθεί η λίστα για να βρεθούμε στο τέλος της.

```
struct forologumenos{
   // να συμπληρωθεί [A]
  string onoma;
  string afm;
  int eisodhma;
  struct forologoumenos *next;
};
struct linked_list {
    struct forologumenos* head = NULL;
    int size=0;
};
```

```
void erotima2(struct linked_list &alist){
  // να συμπληρωθεί [B]
  forologoumenos* cur=alist.head;
  while (cur != NULL){
    if (cur->eisodhma<
10000)
```

	cout << cur->onoma << " " << cur->afm << endl;
	cur = cur->next;
	}
	}

Θέμα 3 [3 μονάδες]

1. Η στατική αναπαράσταση λίστας παρουσιάζει το πλεονέκτημα του άμεσου εντοπισμού ενός στοιχείου της λίστας με βάση τη θέση του. (Σ/Λ) ΣΩΣΤΟ
2. Η δυαδική αναζήτηση μπορεί να εφαρμοστεί μόνο σε ταξινομημένους πίνακες. (Σ/Λ) ΣΩΣΤΟ
3. Η σειριακή αναζήτηση είναι ταχύτερη από την δυαδική αναζήτηση. (Σ/Λ) ΛΑΘΟΣ
4. Ποια είναι η παράμετρος που θα πρέπει να συμπληρωθεί κατά τη μεταγλώττιση του κώδικα, έτσι ώστε να χρησιμοποιηθούν οι δυνατότητες της C++ 11; -std=c++11

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων
Μάθημα: Δομές Δεδομένων και Αλγόριθμοι (Εργαστήριο)
Ακαδημαϊκό έτος 2018-2019
ΠΡΟΟΔΟΣ - ΔΕΚΕΜΒΡΙΟΣ 2018
Διάρκεια εξέτασης: 60 λεπτά
ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Α.Μ.:

ΘΕΜΑΤΑ Β

Θέμα 1 [4 μονάδες]

1. Να συμπληρώσετε τον αλγόριθμο δυαδικής αναζήτησης στα σημεία Α, Β, Γ και Δ. Η συνάρτηση binary_search θα πρέπει να επιστρέφει τη θέση στην οποία βρίσκεται το στοιχείο key στον πίνακα a, λαμβάνοντας υπόψη τα στοιχεία του a από τη θέση aristera μέχρι και τη θέση dexia. Αν το στοιχείο key δεν υπάρχει, η συνάρτηση θα πρέπει να επιστρέφει την τιμή -1.

```
int binary_search(int a[], int aristera, int dexia, int key) {
 int m = (aristera + dexia) / 2;
 if (dexia < aristera) {
   [A] return -1;
 } else if ((key - a[m]) > 0) {
   [B] return binary_search(a, m+1, dexia, key);
 } else if ((key - a[m]) < 0) {
   [Γ] return binary_search(a, aristera, m-1, key);
 }
 else {
   [Δ] return m;
 }
}
```

2. Έστω πίνακας 16 θέσεων που περιέχει τις τιμές {1,2,3,5,7,8,10,12,14,15,16,17,18,20,21,23}. Χρησιμοποιώντας τη δυαδική αναζήτηση, ποια θα είναι τα στοιχεία του πίνακα με τα οποία θα συγκριθεί η τιμή 4 έτσι ώστε ο αλγόριθμος να επιστρέψει -1, δηλώνοντας ότι το συγκεκριμένο κλειδί δεν υπάρχει στον πίνακα;
   Θα συγκριθεί με τις τιμές: 12, 5, 8, 7

3. Γράψτε τον κώδικα που δηλώνει και αρχικοποιεί τον πίνακα του προηγούμενου ερωτήματος, που πραγματοποιεί την κλήση της συνάρτησης για την αναζήτηση της τιμής 4 στον πίνακα και που εμφανίζει το αποτέλεσμα που επιστρέφει η συνάρτηση.

```
int a[]={1,2,3,5,7,8,10,12,14,15,16,17,18,20,21,23};
cout << binary_search(a, 0, 15, 4) << endl;
```

Θέμα 2 [3 μονάδες]

1. Για μια απλά συνδεδεμένη λίστα συμπληρώστε τον κώδικα [A] που περιγράφει έναν κόμβο της (struct node) με ένα στοιχείο δεδομένων με όνομα data και τύπο δεδομένων double.
2. Συμπληρώστε τον κώδικα [B] που επιστρέφει το άθροισμα των πεδίων data για όλα τα στοιχεία της λίστας από την κεφαλή της μέχρι και το στοιχείο που βρίσκεται στην i-οστή θέση της.
3. Σε μια απλά συνδεδεμένη λίστα είναι ταχύτερο να εισάγουμε νέα στοιχεία στην αρχή, ή στο τέλος, ή δεν έχει διαφορά; Αιτιολογήστε την απάντησή σας.
   Είναι ταχύτερο να εισαχθεί στην αρχή καθώς δεν χρειάζεται να διανυθεί η λίστα για να βρεθούμε στο τέλος της.

```
struct node{
    // να συμπληρωθεί [A]
 double data;
 node* next;
};

struct linked_list {
    struct node* head = NULL;
    int size=0;
};
```

```
double sum(struct linked_list &l, int i){
 if (i<0 || i>l.size)
  return NULL;
 // να συμπληρωθεί [B]
 node* current = l.head;
 double s=0.0;
 for(int j=0;j<=i;j++) {
  s += current->data;
  current = current->next;
 }
 return s;
}
```

Θέμα 3 [3 μονάδες]

1. Η αναπαράσταση γραμμικής λίστας με συνδεδεμένη λίστα επιτρέπει την απευθείας μετάβαση σε οποιοδήποτε στοιχείο της λίστας με βάση την τιμή του δείκτη του. (Σ/Λ)
   ΛΑΘΟΣ
2. Η σειριακή αναζήτηση μπορεί να εφαρμοστεί μόνο σε μη ταξινομημένους πίνακες. (Σ/Λ)
   ΛΑΘΟΣ
3. Η δυαδική αναζήτηση μπορεί να εφαρμοστεί μόνο σε ταξινομημένους πίνακες. (Σ/Λ)
   ΣΩΣΤΟ
4. Ποιος είναι ο κώδικας με τον οποίο δεσμεύεται δυναμικά στη μνήμη χώρος για την αποθήκευση N float τιμών είναι και ποιος ο κώδικας με τον οποίο ο χώρος αυτός αποδίδεται πίσω στο σύστημα, στη C++;
   ```
   float *a = new float[N];
   delete [] a;
   ```