ΠΡΟΟΔΟΣ Β'

Α.Μ.:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

## Θέμα 1 [4 μονάδες]

Στο πρόγραμμα stack_oo.cpp του εργαστηρίου 5 προσθέστε μια συνάρτηση με όνομα pop_values που να δέχεται ως παράμετρο έναν ακέραιο αριθμό k. Η συνάρτηση να απωθεί, εφόσον αυτό είναι δυνατόν, k τιμές από τη στοίβα τις οποίες να τις επιστρέφει σε ένα vector. Καλέστε τη συνάρτηση pop_values από τη main. Χρησιμοποιήστε μια στοίβα με μέγιστη χωρητικότητα 100 θέσεων στην οποία εφόσον πρώτα έχετε τοποθετήσει τους 50 πρώτους θετικούς ακεραίους στη συνέχεια να καλείτε τη συνάρτηση pop_values με όρισμα 10 και να εμφανίζετε τις τιμές που επιστρέφει . Παρατήρηση: Γράψτε μόνο τη συνάρτηση pop_values και τη συνάρτηση main.

```
  vector<T> pop_values(int k) {
    if (top - k < 0)
      throw "not enough elements";
    vector<T> v;
    for (int i = 0; i < k; i++) {
      T x = pop();
      v.push_back(x);
    }
    return v;
  }

int main() {
  my_stack<int> astack(100);
  for (int i = 1; i <= 50; i++)
    astack.push(i);
  vector<int> v = astack.pop_values(10);
  astack.print();
  for (int x : v)
    cout << x << " ";
  cout << endl;
}
```

## Θέμα 2 [4 μονάδες]

Γράψτε πρόγραμμα που να ορίζει τη δομή graduate (απόφοιτος) με πεδία name (όνομα) και grade (βαθμό). Στη main ο χρήστης να εισάγει τα στοιχεία 5 αποφοίτων από το πληκτρολόγιο οι οποίοι να εισέρχονται σε μια ουρά προτεραιότητας (std::priority_queue) με όνομα candidates (υποψήφιοι) θεωρώντας ότι προηγείται ο υποψήφιος με τον υψηλότερο βαθμό. Να εμφανίζονται τα περιεχόμενα της ουράς προτεραιότητας.

```
#include <iostream>
#include <queue>

using namespace std;

struct graduate {
        string name;
        double grade;
        bool operator<(const struct graduate& other) const {return grade < other.grade;}
```

```
};

int main() {
	priority_queue<graduate> candidates;
	for (int i = 0; i < 5; i++) {
		graduate x;
		cout << "Enter name: ";
		cin >> x.name;
		cout << "Enter grade: ";
		cin >> x.grade;
		candidates.push(x);
	}

	while (!candidates.empty()) {
		graduate x = candidates.top();
		cout << x.name << " " << x.grade << endl;
		candidates.pop();
	}
}
```

## Θέμα 3 [2 μονάδες]

Ένας σωρός ελαχίστων (MINHEAP) έχει αποθηκευτεί σε ένα πίνακα ως εξής: [2,5,9,6,8,20,10,12,18,11]. Σχεδιάστε το σωρό ως δένδρο. Στο ίδιο σχήμα προσθέστε την τιμή 4. Διαγράψτε τη μικρότερη τιμή του **αρχικού σωρού** (δηλαδή χωρίς την προσθήκη της τιμής 4) και σχεδιάστε εκ νέου το σωρό ως δένδρο.