

1. Γραψετε μια γενική (πρότυπη) **void** συνάρτηση **swapx** η οποία ανταλλάσει το περιεχόμενο (κάνει swap) μεταξύ δυο μεταβλητών του ίδιου τύπου.

2. Γραψετε μια γενική (πρότυπη) **void** συνάρτηση η οποία εμφανίζει στην οθόνη δυο μεταβλητές (οποιοδήποτε τύπου)

3. Δίδεται η τάξης

```
class A {
    public:
    int x;
    float y;
    char c;
    char s[20];
};
```

Στην Άσκηση 1 υπερφορτώσετε την **swapx** με μια συνάρτηση η οποία ανταλλάσει το περιεχόμενο (αποφασίσετε εσείς τα μέλη) δυο αντικειμένων του τύπου A αφού προηγουμένως έχετε δώσει τιμές από το πληκτρολόγιο στα δυο αντικείμενα.

4. Δίδεται η γενική τάξης

```
template <class X, class Y>
class AB {
    X i;
    Y k;
    public:
    AB(X a, Y b);
    X geti() { return(i); };
    Y getk();
};
```

Να υλοποιήσετε τις συναρτήσεις μέλη της γενικής τάξης **AB** και ελέγξετε την ορθότητά της με το παρακάτω κώδικα

```
AB<int, char> ob1(12, 'Z');
cout << ob1.geti() << " " << ob1.getk() << endl;

AB<long, double> ob2(5L, 3.1);
cout << ob2.geti() << " " << ob2.getk() << endl;
```

5. Δίδεται το παρακάτω πρόγραμμα υλοποίησης μιας στοιβάδας τύπου int. Αφού κατανοήσετε την λειτουργία αυτού του προγράμματος, να τροποποιήσετε την στοιβάδα σε γενική ώστε να λειτουργία για κάθε τύπο δεδομένων. Τεστ main για την γενική στοιβάδα δίδεται παρακάτω.

```
class stack {
    private:
        int size;
        int *t, *v;
    public:
        stack(int sz=10);
        ~stack();
        void push (int val);
        int pop();
        inline bool empty() { return v==t; }
};

stack::stack(int sz)
{
    t=v=new int[size=sz];
```

```

}

stack::~~stack()
{
    delete []v;
}

void stack::push(int val)
{
    if (t-v==size) return;
    *t+=val;
}

int stack::pop()
{
    if (t==v) return 0;
    return *--t;
}

int main()
{
    stack si(20);

    si.push(10);
    si.push(20);
    si.push(30);
    si.push(40);
    si.push(50);

    while (!si.empty())
        cout << si.pop() << endl;
}

```

Τεστ main για την γενική στοιβάδα

```

int main()
{
    stack<int> si(20);

    si.push(10);
    si.push(20);
    si.push(30);
    si.push(40);

    while (!si.empty()) cout << si.pop() << endl;

    stack<float> sf(20);

    sf.push(1.1);
    sf.push(2.2);
    sf.push(3.3);
    sf.push(4.4);
    sf.push(5.5);

    stack<float> g;
    cout << "sss" << sizeof(g) << endl;

    while (!sf.empty()) cout << sf.pop() << endl;
}

```