

## **UPPGIFT1** **uppgift1.cpp**

```
#include <iostream>
#include <cstring>

using namespace std;

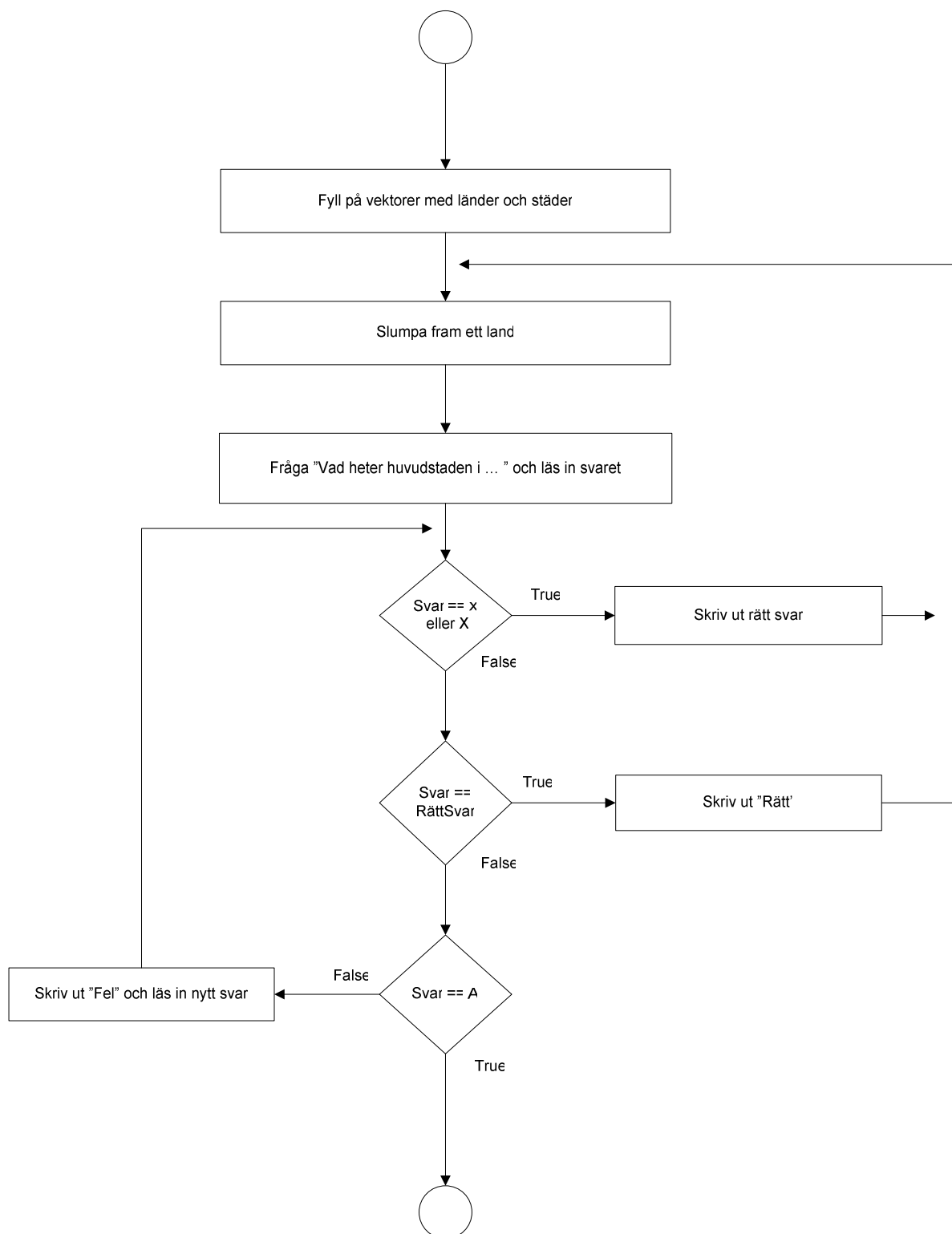
int main()
{
    char losen1[20];
    char losen2[20];
    bool losreg = false;

    do{
        cout << "Ange nytt lösenord [max 20 tkn]: ";
        cin >> losen1;
        cout << "Verifiera lösenord [max 20 ktn]: ";
        cin >> losen2;

        if(strcmp(losen1,losen2)==0)
        {
            cout << "Lösenordet är registrerat" << endl;
            losreg = true;
        }
        else
        {
            cout << "Lösenordet stämmer ej" << endl;
        }
    }while(losreg==false);

    return 0;
}
```

## UPPGIFT2 Diagram



## uppgift2.cpp

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <cctype>
using namespace std;

const int ANTAL_LANDER = 5;
const int ANTAL_BOKSTAVER = 20;
const int SVARSTORLEK = 100;

//Fyller inskickade vektorer med länder och städer.
//Pre: Vektorerna har nog med plats
//Post: Fyllt vektorer med länder och städer

void fyllNamn(char Vektor1[][ANTAL_BOKSTAVER], char Vektor2[][ANTAL_BOKSTAVER])
{
    strcpy(Vektor1[0], "Sverige");
    strcpy(Vektor1[1], "Norge");
    strcpy(Vektor1[2], "Finland");
    strcpy(Vektor1[3], "Island");
    strcpy(Vektor1[4], "Danmark");

    strcpy(Vektor2[0], "Stockholm");
    strcpy(Vektor2[1], "Oslo");
    strcpy(Vektor2[2], "Helsingfors");
    strcpy(Vektor2[3], "Reykjavík");
    strcpy(Vektor2[4], "Köpenhamn");
}

//Genererar ett slumtal som motsvarar ett land
//Pre: Antal > 0
//Post: Levererat ett slumtal mellan 0 och antal länder.

int genLand(int antal)
{
    int slump;
    slump = rand()%antal;

    return slump;
}

//Kontrollerar användarens svar
//Pre: Finnas länder i Vektor1, finnas städer i Vektor2, akt_land >= 0
//Post: Returnerat True om korrekt svar eller False om felaktigt svar, vid x/X som
//svar returnerat true.

bool checkSvar(char Vektor1[][ANTAL_BOKSTAVER], char Vektor2[][ANTAL_BOKSTAVER],
int akt_land, char InSvar[])
{

```

```
bool retry = false;

char RattTemp[ANTAL_BOKSTAVER];

if((strcmp(InSvar,"x") == 0) || (strcmp(InSvar,"X") == 0))
{
    cout << "Huvudstaden i " << Vektor1[akt_land] << " är "
        << Vektor2[akt_land] << endl;
    retry = false;
}
else
{
    for(int i = 0; i < SVARSTORLEK; i++)
    {
        InSvar[i] = toupper(InSvar[i]);
    }
    for(int k = 0; k < ANTAL_BOKSTAVER; k++)
    {
        RattTemp[k] = toupper(Vektor2[akt_land][k]);
    }

    if(strcmp(InSvar,RattTemp) == 0)
    {
        cout << "Rätt svar!" << endl;
        retry = false;
    }
    else
    {
        cout << "Fel svar, försök igen: ";
        retry = true;
        cin >> InSvar;
    }
}

return retry;
}

//Skriver ut fråga om land och tar emot användarens svar
//Pre: Finns länder i Vektor1, index >= 0
//Post: Ändrat innehållet i den variabel som UtSvar refererar till.

void stallFrage(char Vektor1[][ANTAL_BOKSTAVER], int index, char UtSvar[])
{
    cout << "Vad heter huvudstaden i " << Vektor1[index] << "?" << endl
        << "Ange ditt svar: ";
    cin >> UtSvar;
}
}
```

```
int main()
{
    unsigned int aktuellt_land, old_land;
    char svar[SVARSTORLEK];

    char Lander[ANTAL_LANDER][ANTAL_BOKSTAVER];
    char Stader[ANTAL_LANDER][ANTAL_BOKSTAVER];

    srand(time(0));

    //Fyller i vektorer med länder och städer
    fyllNamn(Lander, Stader);

    //Kör så länge användaren inte matat in A
    do
    {
        //Ser till så att inte samma land kommer två gånger efter varandra
        do
        {
            old_land = aktuellt_land;
            aktuellt_land = genLand(ANTAL_LANDER);
        }while(old_land == aktuellt_land);

        //Ställer fråga om land
        stallFraga(Lander, aktuellt_land, svar);

        //Kontrollerar svaret, och repiterar ifall fel svar angivits
        if (strcmp(svar, "A") != 0)
            while((checkSvar(Lander, Stader, aktuellt_land, svar) == true)
                && (strcmp(svar, "A") != 0)) { }

    }while(strcmp(svar, "A") != 0);

    cout << "Tack för att du körde detta program" << endl;

    return 0;
}
```

## **UPPGIFT3**

### **Produktbeskrivning**

Programmet lagrar information om läskedrycker. Det innehåller information om namn, ursprungsland, pris samt volym.

Genom programmet kan man lägga till läskedrycker till listan över de drycker som finns. Att lägga till drycker är möjligt för samtliga användare.

Det finns även en möjlighet att visa alla läskedrycker som ligger inlagda i programmet. Dessa visas i en lista tillsammans med ett ID-nummer. Detta ID-nummer kan man använda för att hämta ut ytterligare specificerad information om drycken.

Innan programmet avslutas så kommer användaren bli tillfrågad om denne verkligen vill avsluta, om denne vill avsluta anger hon ett stort 'J' och trycker RETUR. Detta medför att programmet avslutas och den lagrade listan med läskedrycker raderas från minnet.

## uppgift3.cpp

```
#include <iostream>
#include <iomanip>
using namespace std;

const int ANT_TKN = 100;
const int MAX_LASK = 30;

struct lask
{
    char namn[ANT_TKN];
    char land[ANT_TKN];
    float pris;
    int mangd;
};

//Ser till att inmatat värde endast kan anta siffra
//Pre: True
//Post: Returnerat integer

int mataIn()
{
    char buffer[100];
    int varde;
    bool felKoll = false;
    do
    {
        if(felKoll == true)
            cout << "Felaktigt värde! Försök igen: ";
        cin.getline(buffer, 100);
        if(buffer[0]!='0')
            return 0;
        felKoll = true;
    }while(!(varde=atoi(buffer)));
    return varde;
}

//Lägger till Läsk i vektor
//Pre: Vektor har ledig plats och ant >= 0
//Post: Läsk har adderats till vektor
void lagg_till(lask AllaSorter[], int & ant)
{
    if(ant < MAX_LASK)
    {
        cout << "LÄGG TILL NY SORT" << endl
            << "=====" << endl << endl
            << "Ange sortens namn: ";

        cin >> AllaSorter[ant].namn;

        cout << "Ange ursprungsland: ";

        cin >> AllaSorter[ant].land;
```

```
    cout << "Ange sortens pris (kr): ";

    cin.get();

    AllaSorter[ant].pris = mataIn();

    while(AllaSorter[ant].pris <= 0)
    {
        cout << "Felaktigt värde! Försök igen: " ;
        AllaSorter[ant].pris = mataIn();
    }

    cout << "Ange sortens mängd (cl): ";

    AllaSorter[ant].mangd = mataIn();

    while(AllaSorter[ant].mangd <= 0)
    {
        cout << "Felaktigt värde! Försök igen: " ;
        AllaSorter[ant].mangd = mataIn();
    }

    cout << endl << "Sorten lagrad";

    ant ++;
}
else
    cout << "Du kan inte lägga till fler artiklar!";

}

//Visar nummer och namn för all läsk i lista
//Pre: Det finns antal ant poster i AllaSorter
//Post: Skrivit ut lista på samtliga läsker

void lista_sort(lask AllaSorter[], int ant)
{
    cout << "LISTA ALLA LÄSKSORTER" << endl
        << "======" << endl;

    cout << "Nummer" << "\t" << "Namn" << endl;

    for(int i = 0; i<ant; i++)
        cout << i << "\t" << AllaSorter[i].namn << endl;

    if(ant <= 0)
        cout << endl << " ** Det finns inga artiklar i listan ** ";

    cout << endl << endl;
}
}
```



```
//Skriver ut information om specefik läsk
//Pre: Det finns poster i AllaSorter
//Post: Har skrivit ut information om läsk.

void skriv_info(lask AllaSorter[], int antal)
{
    int id;

    cout << "SKRIV UT INFORMATION OM LÄSK" << endl
         << "======" << endl << endl;

    if(antal > 0)
    {
        cout << "Ange läskens ID-nummer [ange -1 för att avbryta]: ";

        id = mataIn();

        while((id < -1) || (id > antal))
        {
            cout << "Detta värde är ej giltigt! Ange nytt: ";
            id = mataIn();
        }

        if(id != -1)
        {
            cout << "*****" << endl << endl;

            cout << "Namn: " << AllaSorter[id].namn << endl
                 << "Ursprungsland: " << AllaSorter[id].land << endl
                 << "Pris (kr): " << AllaSorter[id].pris << endl
                 << "Mängd (cl): " << AllaSorter[id].mangd << endl;

            cout << endl << endl;
        }
    }
    else
        cout << "*** Det finns inga artiklar i listan ***" << endl;
}

//Kontrollerar om användaren vill avsluta
//Pre: True
//Post: Returnerat true om användaren vill avsluta, annars false.

bool kolla_avsluta()
{
    char stop;
    bool sluta = false;

    cout << "Är du säker på att du vill avsluta? (J/N) ";

    cin >> stop;

    if(stop == 'J')
        sluta = true;
}
```

```
cin.get();

return sluta;

}

//Skriver ut programmets meny samt tar emot val
//Pre: True
//Post: Returnerat användarens val
int skriv_meny()
{
    int val;
    cout << endl;
    cout << "LÄSKPROGRAMMET" << endl
         << "=====" << endl << endl
         << "1. Lägg till läsk" << endl
         << "2. Lista alla läsksorter" << endl
         << "3. Skriv ut information om läsk (id krävs)" << endl
         << "0. Avsluta" << endl << endl
         << "Ange val: ";

    val = mataIn();

    return val;
}

int main()
{
    lask sorter[MAX_LASK];
    int antal_lagrade = 0;
    bool avsluta = false;
    int meny_val;

do
    {
        meny_val = skriv_meny();

        switch(meny_val)
        {
            case 1: lagg_till(sorter, antal_lagrade); break;
            case 2: lista_sort(sorter, antal_lagrade); break;
            case 3: skriv_info(sorter, antal_lagrade); break;
            case 0: avsluta = kolla_avsluta(); break;
            default: cout << "Felaktig inmatning" << endl;
        }
    }while(avsluta != true);

    return 0;
}
```

```
int main()
{
    lask sorter[MAX_LASK];
    int antal_lagrade = 0;
    bool avsluta = false;
    int meny_val;

do
{
    meny_val = skriv_meny();

    switch(meny_val)
    {
        case 1: lagg_till(sorter, antal_lagrade); break;
        case 2: lista_sort(sorter, antal_lagrade); break;
        case 3: skriv_info(sorter); break;
        case 0: avsluta = kolla_avsluta(); break;
        default: cout << "Felaktig inmatning" << endl;
    }
}while(avsluta != true);

    return 0;
}
```