



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 1	Poäng 1	Utgåva 3	Datum 2012-12-30	Framtagen av Na/Lö

Om och om igen

OBLIGATORISK

Problem

Skriv ett program som läser in ett godtyckligt antal heltal, tills användaren väljer att avsluta inmatningen. Talen ska lagras i en *vektor*¹ som *allokeras dynamiskt* (d.v.s. en pekare till ett vektorutrymme). Användaren ska i en repetition kunna välja om ett nytt tal ska läsas in och programmet ska då allokera om, dvs. utöka utrymmet ett element i taget för vektorn, enligt följande grova algoritm:

```
initiera antal till 0 och en pekare till en heltalsvektor till nullptr.  
läs tal  
så länge tal inte är stopptal  
    reservera en temporär vektor som är ett element större än den gamla  
    kopiera samtliga värden från den gamla vektorn till den nya. Det behövs inga specialfall för första talet.  
    frigör utrymmet som den gamla pekaren pekade på.  
    tilldela pekaren (talvektorn) adressvärdet för den nya vektorn  
    spara ditt tal i den nya större vektorn  
    öka antal  
    läs tal  
end  
skriv ut talen  
avallokera vektorn och nollställ denna
```

Testfall

```
Enter integer numbers to store <Cancel with 'Stop'>:  
23 12 -3 4  
55 3 21 2 4  
23 stop  
23 12 -3 4 55 3 21 2 4 23  
Press any key to continue . . .
```

Tips

Normalt sett utökas inte ett dynamiskt allokert utrymme med bara 1 element, utan oftare med en viss % -sats. Dock gör vi undantagsvis så här i denna uppgift, bara för att lära oss principerna. Du som vill, kan fördjupa dig om ovanstående på kurshemsidan under *Fördjupningsmaterial*. Där finns ett exempel som utnyttjar en (relativt avancerad) funktion för ändamålet, vilken inte ska användas i denna uppgift!

När du använder char eller string-typ som stopp-värde, kan du dra nytta av cin-objektets beteende vid läs-operationer. Studera följande exempel på while-loop, där *nr* är av int-typ:

```
while (cin >> nr) // Så länge cin läser numeriska värden, kommer läs-operationen att resultera i värdet true.  
{ // Om icke numeriska värden matas in (exempelvis "Stop" eller "Q") kommer inläsningen att misslyckas.  
    // Detta resulterar i returvärdet false, varvid loopen bryts!  
    // Om fler inläsningar ska göras här efter, måste cin-objektets felstatus återställas.  
    // För tips på hur detta görs – läs dokument nr. 5. "Felhantering", på kurswebbsidan "Tips och verktyg".
```

OBS! Skapa inget utrymme förrän det behövs. Man kan starta med en "tom" pekare och när ett nytt tal har lästs in, skapar du utrymme för talet och sparar det.

¹ En endimensionell array kallas ofta för en vektor.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 2	Poäng 1	Utgåva 2	Datum 2009-09-30	Framtagen av Lö

Vad är det för fel?

Problem

Hittade följande fråga i en newsgroup! Vad är det för fel? Hur rättar man det? (Felet är inte omfattande)

"Why does my program always turn up "heads were rolled 10 times, tails were rolled 0 times" or vice versa? I'm trying to write a coin tossing program that tosses a coin 10 times and reports how many times heads were rolled and how many times tails were rolled. Right now there's something I'm doing wrong (beginner you know) that is causing the outcome to be pretty unlikely. Here is my program":

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

enum Coin {Tails, Heads};

int main()
{
    int frequency0=0, frequency1 = 0;
    Coin face;

    for (int counter = 1; counter <= 10; ++counter)
    {
        srand(time(0));
        face = Coin(rand()% 2);

        switch (face)
        {
            case Tails:    ++frequency0;
                          break;
            case Heads:    ++frequency1;
                          break;
        }
    }
    cout << "Tails were rolled " << frequency0 << " times" << endl;
    cout << "Heads were rolled " << frequency1 << " times" << endl;

    cin.get();
    return 0;
}
```

Redovisning

Beskriv vad som behöver ändras för att programmet ska fungera som tänkt (kan göras som kommentarer i den redovisade koden). Justera också lösningen så att den fungerar på det förväntade sättet.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 3a	Poäng 1	Utgåva 2	Datum 2012-12-18	Framtagen av Na/Lö

Små mysterier

Problem

Denna uppgift innehåller spelkortssymboler, vilka kanske bara finns i Windows, men uppgiften får då lösas lite annorlunda med mindre ändringar.

- Skriv ett litet program som multiplicerar ett inmatat heltal med 4, men utan att använda några *aritmetiska* operatorer!
- Anta att du skapar ett program som hanterar strukturer (se nedanstående kodutdrag). För en viss programuppgift behöver man veta hur stort utrymme varje strukturpost tar. Det är alltid riskfyllt att manuellt beräkna eller göra antaganden om typers storlek (vilket kan vara olika i olika system). Du ska därför låta programmet göra den korrekta beräkningen av detta. Hur ser koden ut som utför detta, och ger resultatet i antal byte?

```
struct StudentTyp
{
    char persNr[12];
    char namn[50];
    int arskurs;
    bool kurser[40];
    double result[50];
};
```

```
StudentTyp student = {"", "", 0, {0}, {0}};
```

```
int storlek = ...?           // Vad saknas, och vad skrivs ut på skärmen?
cout << storlek << endl;
```

- Komplettera nedanstående lilla program med en *enumeration*. D.v.s. du ska skriva den inledande *enum*-deklaration som saknas, och initiera den på sådant sätt, så att programmet skriver ut de fyra valörerna i en kortlek på skärmen, enligt följande bild:



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

```
// Vad saknas ...?
```

```
int main()
{
    Card h = Hjarter, r = Ruter, k = Klover, s = Spader;

    cout << char(h) << " ";
    cout << char(r) << " ";
    cout << char(k) << " ";
    cout << char(s) << "\n";

    return 0;
}
```



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 3a	Poäng 1	Utgåva 2	Datum 2012-12-18	Framtagen av Na/Lö

Tips

För a) är vänsterskiftoperatören (kursboken s. 202) av intresse för att lösa uppgiften. För b) har du användning av en operator, som introduceras i kursbokens teori för Steg 5. Läs om *enumerationer*, för att lösa c).

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 3b	Poäng 1	Utgåva 2	Datum 2009-09-30	Framtagen av Lö

Pekproblem

Problem

Din uppgift blir här att i ord beskriva och förklara vad de olika kommenterade kodavsnitten 1-4 gör i respektive uppgift.

Uppgift A

```
//1. Do something
int *arr = new int[10];

int *iptr = arr+9;
int i = 9;
while (iptr>=arr)
{
    *iptr-- = i--;
}

//2. Do something else
for (int i=0; i<10; ++i)
    cout << setw(2) << arr[i];
cout << endl;

//3. Finally
iptr = arr + 2;
cout << *(iptr+1);

//4. Wrap up
delete[] arr;
arr = NULL;
```

Uppgift B.

```
char str[] = "ABC";
int len = strlen(str);

//1. Do something
char *cpy = new char[len+1];
strcpy(cpy, str, len+1);

//2. Do something else
char *cPtr = cpy;
while (*cPtr)
{
    *cPtr += 1;
    ++cPtr;
}

//3. Finally
cout << " 1. " << str
      << " 2. " << cpy << endl;

//4. wrap up
delete[] cpy;
cpy = NULL;
```

Testfall

Tips

Det är inte nödvändigt att skriva in och provköra koden. Det går att lösa teoretiskt.

Redovisning

Du ska (i teoretiska ordalag) beskriva vad som utförs i de kommenterade kodavsnitten 1-4 i respektive uppgift. Om du vill går det bra att göra detta i form av kommentarer inuti koden.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/DFM
Steg 5	Uppgift 4	Poäng 2	Utgåva 3	Datum 2012-12-18	Framtagen av Lö

Shuffle

Problem

En mycket vanlig uppgift för en dator är att sortera element i stigande eller fallande ordning. Motsatsen är också vanligt, d.v.s. att skapa oordning, eller helt enkelt att blanda elementen. Det kan vara en startlista som ska tas fram, en kortlek som ska blandas, etc.

Du ska skriva ett program som *blandar ett godtyckligt antal heltal som finns i en dynamiskt allokerad vektor*¹. Misslyckad allokering ska medföra att programmet avslutas. (Använd try – catch – se exempel i steg 5). Vektorn ska innan den blandas innehålla talen 0 - (n - 1), där n är antalet element.

Indata ska vara antalet tal, n, i vektorn – samt validerat att n är >= 0.

Utdata ska vara utskrift av först den oblandade vektorn och sedan den blandade vektorn. Skriv ut vektorn snyggt i tabellform med ett visst antal angivna kolumner.

Lägg upp programmet med 3 funktioner som:

- skapar den ursprungliga vektorn: `void fyll(int *vek, int n);`
- blandar talen i vektorn: `void blanda(int *vek, int n);`
- skriver ut vektorn: `void skrivUt(const int *vek, int n, int k);`

Parametern **vek** pekar på en vektor med n stycken element. För att hindra ändring av de element som **vek** pekar på är **const**-deklarerats denna. **k** är antal kolumner vid utskrift.

En algoritm för att blanda en vektor kan se ut enligt nedan:

Så länge som n > 1 så:

Slumpa ett index mellan 0 och n - 1.

Byt elementet i det slumpade indexet med elementet på index n - 1 (det sista i vektorn första gången!)

Minska n med 1

Testfall

Körexempel

```
Ange antal heltal att blanda: -10
Du maste ange positivt heltal?
Ange antal heltal att blanda: 25
Oblandade tal:
 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24
Blandade tal:
21  8 18 13 22 15 19 10  7  1
 6  3 23 14  5 24 20  0  2  4
17 12  9 11 16
En gang till? <J/N> :
```

¹ En endimensionell array kallas ofta för en vektor.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 5	Poäng 2	Utgåva 3	Datum 2012-12-18	Framtagen av Lö

Gissa talet

Problem

Följande är en gissningslek mot datorn. *Funktionen* "test" används för att undersöka och därefter meddela resultatet av gissningen. Komplettera koden med det som saknas för att gissningsleken ska fungera.

Det finns ett sätt att gissa, som gör att du alltid kan vinna mot datorn – kommer du på hur? (Det kallas binärsökning! ☺)

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

const int MAXTRY = 7;
enum Utfall {TOOLOW, TOOHIGH, OK};

// Prototyper
void spelomgang();
```

// 1.Skriv den saknade funktionsdeklarationen här...

```
int main()
{
    srand((int)(time(0)));
    char svar;

    do
    {
        system("CLS");
        cout << "Spelet HI-LO" << endl
              << "*****" << endl << endl;
        spelomgang();
        cout << "En gang till (J/N)? ";
        cin >> svar;
    }while (svar == 'J' || svar == 'j');

    return 0;
}

void spelomgang()
{
    int tal;
    bool klar = false;
    int hemligt = 1 + rand() % 100;
    int antalForsok = 0;

    // Flagga för spelstatus
    // slumpa fram det hemliga talet

    do
    {
        cout << "Gissa ett tal mellan 1-100: ";
        cin >> tal;
        ++antalForsok;
        switch (test(tal, hemligt))
        {
```



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 5	Poäng 2	Utgåva 3	Datum 2012-12-18	Framtagen av Lö

```
case TOOLOW : cout << "For litet! ";
               break;
case TOOHIGH : cout << "For stort! ";
                break;
case OK      : cout << "Du gissade rätt på "
                  << antalForsok << " forsok!" << endl;
                klar = true;
                break;
            }
    if (!klar)
    {
        cout << MAXTRY - antalForsok << " forsok kvar!\n";
    }
} while (!klar && antalForsok<MAXTRY);

if (!klar) //Om spelet avslutades på MAXTRY försök
{
    cout << "Tyvarr, du forlorade...:-(\n. Talet var " << hemligt << endl;
}
}
```

// 2. Skriv funktionsdefinitionen här

```
{
    /* Funktionen jämför det gissade talet med det slumpade och returnerar enumvärdena TOOLOW,
    TOOHIGH el.OK*/
}
```

Testfall

Körexempel

```
Spelet HI-LO
*****

Gissa ett tal mellan 1-100: 50
For litet! 6 forsok kvar!

Gissa ett tal mellan 1-100: 76
For stort! 5 forsok kvar!

Gissa ett tal mellan 1-100: 63
For stort! 4 forsok kvar!

Gissa ett tal mellan 1-100: ... osv.
```

Tips

Vad som förväntas ske i programmet, bör du kunna tolka ut genom att noga analysera befintlig kod (vilken inte får ändras!).
Tänk på att den efterfrågade funktionen ska returnera en datatyp som passar switch-satsen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++			Institution Lnu/DFM	
Steg 5	Uppgift 6	Poäng 1-3	Utgåva 3	Datum 2012-12-19	Framtagen av Na/Lö

Glasskiosken

Denna tillämpning är upplagd som en "3-stegsuppgift", vilket innebär följande:

För varje deluppgift a) – c) som du genomför, får du 1 poäng. Det är valfritt att avsluta arbetet efter resp. steg och därmed få motsvarande poängtal. Uppgifterna ska dock utföras i ordningen a – c, eftersom varje efterföljande programdel bygger på den föregående.

Problem

Strukturer hanteras ofta – liksom vektorer och strängar – som argument till funktioner. När dessa blir stora, är det mest praktiskt att skicka en pekare (alternativt referens) till strukturen. Följande program använder en vektor för att lagra strukturer och pekare som argument till funktioner för inläsning resp. utskrift av strukturdata i vektorn.

Se instruktioner för de olika deluppgifterna på följande sida. Komplettera med det som saknas i koden, enligt kursiverade kommentarer. Observera att övrig kod ska vara opåverkad – endast tillägg får göras!

```
#include <iostream>
using namespace std;
const int STR_SIZE = 30;

struct GlassTyp
{
    char tillverkare[STR_SIZE];
    char namn[STR_SIZE];
    int pris; // För enkelhets skull används heltal (kronor)
};

void bestall(GlassTyp *pNyVara); // Läser in data till struktur
GlassTyp handla(const GlassTyp *pKiosk); // Presenterar sortiment och
// Returnerar användarens val

const int ANTAL = 3;

int main()
{
    GlassTyp glass = {"", "", 0};
    GlassTyp kiosken[ANTAL];

    cout << "Beställning av nya glassprodukter till kiosken\n";

    for (int i = 0; i < ANTAL; ++i)
    {
        bestall(&kiosken[i]); // a) Läs in data till varje struktur
    }

    glass = handla(kiosken); // b) Returnera en "glass" som användaren valt

    cout << "\nDu kopte en " << glass.namn << " tillverkad av "
         << glass.tillverkare << " som kostar " << glass.pris << endl;

    cin.get();cin.get(); // Vänta på tangent

    return 0;
}
```



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 6	Poäng 1-3	Utgåva 3	Datum 2012-12-19	Framtagen av Na/Lö

Deluppgift a)

Skriv funktionen nedan, som läser in strukturdata till en GlassTyp-struktur.

```
void bestall(GlassTyp *pNyVara)
{
    // SKRIV funktionskoden med förutsättningar enligt givet funktionshuvud!
}
```

Deluppgift b)

Skriv funktionen nedan, som presenterar "kioskens" (vektorns) sortiment och returnerar användarens "glass"-val, d.v.s. aktuellt element i vektorn.

```
GlassTyp handla(const GlassTyp *pKiosk)
{
    // SKRIV funktionskoden med förutsättningar enligt givet funktionshuvud!
}
```

Deluppgift c)

Komplettera deluppgift b) med den kod som behövs för att garantera en fullgod *felhantering*! Felhantering ska finnas för om användaren matar in något annat än förväntat värde. Exempelvis ska programmet kunna hantera om användaren skriver en bokstav istället för en siffra när det senare krävs, eller om ett valt värde ligger utanför givna alternativ. Felhanteringen måste även fungera för upprepade misstag från användarens sida!

Testfall

...

Tips

Använd pil-operatoren (->) när du läser in data till den struktur som pekaren pekar på.

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++			Institution Lnu/DFM	
Steg 5	Uppgift 5	Poäng 2	Utgåva 2	Datum 2010-01-11	Framtagen av Lö

Gissa talet

Problem

Följande är en gissningslek mot datorn. *Funktionen* "test" används för att undersöka och därefter meddela resultatet av gissningen. Komplettera koden med det som saknas för att gissningsleken ska fungera.

Det finns ett sätt att gissa, som gör att du alltid kan vinna mot datorn – kommer du på hur? (Det kallas binärsökning! ☺)

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

```
const int MAXTRY = 7;
enum utfall {TOOLOW, TOOHIGH, OK};
```

```
//Prototyper
void spelomgang();
```

// 1.Skriv den saknade funktionsdeklarationen här...

```
int main()
{
    srand(int(time(0)));
    char svar;

    do
    {
        system("CLS");
        cout << "Spelet HI-LO" << endl
              << "*****" << endl << endl;
        spelomgang();
        cout << "En gang till (J/N)? ";
        cin >> svar;
    }while (svar == 'J' || svar == 'j');

    return 0;
}

void spelomgang()
{
    int tal;
    bool klar = false;
    int hemligt = 1 + rand() % 100;
    int antalForsok =0;

    // Flagga för spelstatus
    // slumpa fram det hemliga talet

    do
    {
        cout << "Gissa ett tal mellan 1-100: ";
        cin >> tal;
        ++antalForsok;
        switch (test(tal, hemligt))
        {
```



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 5	Poäng 2	Utgåva 2	Datum 2010-01-11	Framtagen av Lö

```
case TOOLOW : cout << "For litet! ";
               break;
case TOOHIGH : cout << "For stort! ";
                break;
case OK      : cout << "Du gissade rätt på "
                  << antalForsok << " forsok!" << endl;
                klar = true;
                break;
            }
    if (!klar)
    {
        cout << MAXTRY - antalForsok << " forsok kvar!\n";
    }
} while (!klar && antalForsok<MAXTRY);

if (!klar) //Om spelet avslutades på MAXTRY försök
{
    cout << "Tyvarr, du forlorade...-(\n. Talet var " << hemligt << endl;
}
}
```

// 2. Skriv funktionsdefinitionen här

```
{
    // Funktionen jämför det gissade talet med det slumpade och skriver
    // ut vägledning för nästa gissning (studera körexempel och main, för att
    // avgöra vilken utskrift som görs av huvudprogram). Antal försök räknas
    // ner och SANT returneras om rätt tal är gissat, annars FALSKT.
}
```

Testfall

Körexempel

```
Spelet HI-LO
*****

Gissa ett tal mellan 1-100: 50
For litet! 6 forsok kvar!

Gissa ett tal mellan 1-100: 76
For stort! 5 forsok kvar!

Gissa ett tal mellan 1-100: 63
For stort! 4 forsok kvar!

Gissa ett tal mellan 1-100: ... osv.
```

Tips

Vad som förväntas ske i programmet, bör du kunna tolka ut genom att noga analysera befintlig kod (vilken inte får ändras!).
Tänk på att den efterfrågade funktionen ska returnera en datatyp som passar switch-satsen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++			Kurs Lnu/DFM	
Steg 5	Uppgift 7	Poäng 2	Utgåva 2	Datum 2012-12-18	Framtagen av Lö

Tidstrukturer

Problem

Du ska skriva ett program som testar lite strukturhantering enligt nedanstående beskrivningar. Huvudprogrammet behöver inte vara så meningsfullt, det ska bara testa de funktioner som du skrivit. Använd följande struktur till uppgifterna.

```
struct TidTyp
{
    int tim;
    int min;
};
```

- a) Skriv en funktion som, med hjälp av en medskickad tidpunkt (i form av ovanstående struktur), beräknar det sammanlagda antalet minuter. Funktionen skall heta `tillMinuter`, och skall kunna användas enligt nedanstående exempel.

```
int main()
{
    int minuter;
    TidTyp tidA = {10,15};
    minuter = tillMinuter(tidA);
    cout << minuter << " minuter" << endl; // Ska skriva: 615 minuter
    return 0;
}
```

- b) Skriv en funktion som omvandlar ett antal minuter, skickade som argument, till ett klockslag i form av ovanstående struktur. Funktionen som skall heta `tillTid`, ska kunna användas enligt nedanstående programexempel. Testa i samma program som uppg. a.

```
int minuter = 124;
TidTyp klockSlag = tillTid(minuter);
```

Därefter ska du skriva ut klockslaget! Med ovanstående data ska det bli: **02:04**

- c) Skriv en funktion, `tidsDifferens`, som beräknar skillnaden mellan två medskickade tidpunkter. Funktionen ska använda ovanstående funktioner, och skall kunna användas enligt nedanstående exempel. (Det är inget som säger att `tid2` ska ha större värde än `tid1`)

```
TidTyp tidB = {10, 30}, tidC = {13, 20}, differens;
differens = tidsDifferens( tidB, tidC);
```

Därefter ska du skriva ut klockslaget så att det med ovanstående data blir: **02:50**

- d) Skriv en funktion, `dynamiskaTider`, som skapar en dynamisk vektor som heter `tider` med plats för 200 tider. (Felhantering behövs inte.) Skriv satser som slumpar in värden till de 200 tiderna. Därefter skriver du kod som beräknar den största tidsdifferensen som finns bland de slumpade talen. Du har nytta av ovanstående funktioner för detta. Avslutningsvis ska vektorn `avallokeras`. Funktionen ska kunna anropas så här:

```
TidTyp maxdiff = dynamiskaTider();
```

`max diff` skrivs därefter ut på samma sätt som uppgift c



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/DFM
Steg 5	Uppgift 7	Poäng 2	Utgåva 2	Datum 2012-12-18	Framtagen av Lö

Testfall

...

Tips

För att enkelt skriva ut inledande 0'or i en tid kan du använda `cout.setfill('0')` tillsammans med `setw(2)`.

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 8	Poäng 2	Utgåva 2	Datum 2009-10-01	Framtagen av Lö

Sten Sax eller Påse

Problem

"Sten, Sax och Påse" är ett spel där två deltagande spelare, samtidigt, väljer något av alternativen sten, sax eller påse. Beroende på vilka val som görs, vinner någon av spelarna enligt följande

- Sten vinner över sax
- Sax vinner över påse
- Påse vinner över sten
- Övriga fall gör spelet oavgjort

Uppgiften blir att skriva ett program som låter en användare spela Sten, Sax och Påse med datorn som automatisk motspelare. Spelaren väljer "Sten", "Sax" eller "Påse" genom att skriva 0, 1 eller 2. Datorn ska därefter göra ett slumpmässigt val. Därefter visas datorns drag, resultatet och poängställningen. Den som först får 10 vinster, vinner hela spelet. Ett krav är att värdena för utfallen Sten, Sax och Påse ska implementeras med *enum*.

Testfall

```
STEN SAX PASE
=====
Ditt drag (Sten=0/Sax=1/Pase=2) : 1
Datorn drar Sten
Datorn vann draget
Poangställning (du-datorn): 0 - 1
Ditt drag (Sten=0/Sax=1/Pase=2) : 2
Datorn drar Pase
Oavgjort
. . .
Poangställning (du-datorn): 9 - 3
Ditt drag (Sten=0/Sax=1/Pase=2) : 0
Datorn drar Sax
Du vann draget
Poangställning (du-datorn): 10 - 3
Du vann med 10-3
En gang till (J/N)
```

Tips

Enumvärden kan inte läsas in. Läs in som *int* och typa till enum-typen. Datorns drag kan också typas till enum-typen.

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/DFM
Steg 5	Uppgift 9	Poäng 3-4	Utgåva 2	Datum 2012-12-18	Framtagen av Lö

Game Of Life

"Game of life" är ett simuleringsspel som uppvisar analogier med de förändringar som äger rum i samhällen med levande individer.

Som spelplan använder vi här en kvadrat med 17*17 rutor. I varje ruta placeras ingen eller högst en individ enligt data som matas in vid simuleringens början. En ruta bestäms med sitt radnummer och kolumnnummer, t ex 4 13 = rad 4, kolumn 13. Från denna första "generation" skall samhället ändra struktur enligt vissa regler för födelse och död.

Nedanstående 3 regler gäller när en generation bildas. Alla förändringar vid en generationsväxling sker samtidigt över hela spelplanen.

1. **Fortlevnad.** En individ lever vidare till nästa generation om den har 2 eller 3 grannar.
2. **Dödsfall.** En individ med 4 eller fler grannar dör av överbefolkning. En individ med 0 eller 1 granne dör av isolering.
3. **Födelse.** I en tom ruta angränsande till exakt 3 individer föds vid nästa generationsväxling en individ.

Man kan förutsätta att det i den yttersta ramen av spelplanen aldrig hamnar någon individ. De tre reglerna behöver därför bara användas på de övriga 15*15 rutorna. Observera att varje sådan ruta har 8 grannrutor (även de diagonalt intilliggande rutorna anses som grannrutor).

Skriv ett program som läser in den första generationen och visar denna på skärmen. De kommande generationerna ska genereras av programmet och visas på skärmen. Användaren bestämmer själv när han vill sluta.

Prototypen använde följande huvudprogram och definitioner. Prototyper till funktionerna och "#include - satser saknas.

```
enum Cell {Unoccupied, Occupied};

const int ROWS = 16;
const int COLS = 16; // Max usable size is 15 x 15
const char ESC = 27;

typedef Cell genType[ROWS+1][COLS+1]; // Row 0 and 16 and col 0 and 16 become a frame around the world

int main()
{
    genType gen; // Matrix to hold a generation
    do // Repeat simulations
    {
        int genNb = 0;

        initGame(gen, genNb); // Init gen to "Unoccupied", genNb to 0 and read startgeneration
        showGeneration(gen, genNb); // Show current generation and genNb
        cout << "\n\nPress any key - ESC stops\n";
        while (_getch() != ESC) // Repeat generations
        {
            nextGeneration(gen, genNb); // Create a new generation and increment genNb
            showGeneration(gen, genNb); // Show current generation and genNb

            cout << "\n\nPress any key - ESC stops\n";
        }
        cout << "Another simulation ? (Y/N) ";
    }while (toupper(_getch()) != 'N'); // _getch() is declared in <conio.h> (Windows only)
    return 0; // Rewrite in other environments
}
```




Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/DFM
Steg 5	Uppgift 9	Poäng 3-4	Utgåva 2	Datum 2012-12-18	Framtagen av Lö

Testfall

Exempel på indata (0 0 används som stoppvärde):

Test 1: 6 9 6 10 7 9 7 10 8 7 8 8 9 7 9 8 0 0
Test 2: 7 7 7 9 8 7 8 8 8 9 9 7 9 9 0 0
Test 3: 8 5 8 6 8 7 8 8 8 9 8 10 8 11 0 0

Exempel på utskrifter från programmet med indata enligt test 3.

Generation 0 	Generation 7 	Generation 8 	Generation 9
Generation 10 	Generation 11 	Generation 12 	Generation 13

Tips

I funktionen *nextGeneration* kan du skapa en temporär generation så att du kan läsa i den aktuella generationen *gen* och skriva till den temporära. Kopiera därefter över den temporära generationen till den aktuella.

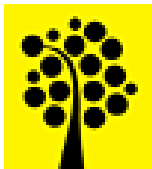
Det är lämpligt att skriva en funktion *grannar* som tar en koordinat och en generation som argument och räknar antalet grannar runt den koordinaten.

Här får du en länk som förklarar life lite mer noggrant + lite större simuleringsmöjligheter:
<http://www.math.com/students/wonders/life/life.html>

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 10	Poäng 3	Utgåva 1	Datum 2012-12-20	Framtagen av Na/Lö

Vän av ordning

Problem

Nedanstående program ber användaren namnge ett antal av sina vänner, sorterar namnen i bokstavsordning, och skriver slutligen ut alla vännernas namn i sorterad ordning.

Komplettera det påbörjade programmet nedan, enligt instruktioner som finns kommenterad i koden. Du ska göra en ¹⁾*sorteringsfunktion* för strängar och en ²⁾*utskriftsfunktion* som skriver ut strängarna. För korrekt minneshantering, ska även en ³⁾*avslutningsfunktion* skrivas, som ser till att allt minne som allokerats återlämnas till systemet. För programlösningen ska följande krav gälla:

Minnesutrymme ska allokeras dynamiskt för inlästa namn. Strängarna (d.v.s. char-pekare) ska lagras i en vektor, som skickas som parameter till resp. funktion. En konstant ska definieras, som anger det ANTAL namn som vektorn rymmer. Observera att ingen befintlig kod nedan får ändras, endast tillägg göras!

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

const int BUFLen = 100;    // Maximal längd på inmatningsbuffert

void sortera(char *kompisar[], int n);    // n är antalet element
void skrivUt(char *kompisar[], int n);    // n är antalet element
void avsluta(char *kompisar[], int n);    // n är antalet element

const int ANTAL = 5;

int main()
{
    char* vanner[ANTAL];    // Vektor med ANTAL st strängpekare
    char buff[BUFLen] = {""};    // Skapar strängbuffert (nullterminerad)
    int antal = 0;

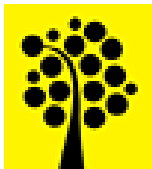
    while (antal < ANTAL)    // Så länge plats finns i vektorn
    {
        cout << "Namnge en vän: ";
        cin.getline(buff, BUFLen);    // Temporär inläsning i strängbuffert

        vanner[antal] = ... // SKRIV KODEN som allokerar minne för strängen

        // SKRIV KODEN som lägger till inläst namn på aktuell plats i vektorn

        ++antal;
    }
    sortera(vanner, antal);    // Sorterar de 'antal' första elementen i vektorn med strängpekare
    skrivUt(vanner, antal);    // Skriver ut de 'anta'l första namnen
    avsluta(vanner, antal);    // Avallokerar allt minne

    return 0;
}
```



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Institution Lnu/DFM
Steg 5	Uppgift 10	Poäng 3	Utgåva 1	Datum 2012-12-20	Framtagen av Na/Lö

```
void sortera(char *kompisar[], int n)
{
    // SKRIV FUNKTIONEN som sorterar vektorns strängar i bokstavsordning!
}

void skrivUt(char *kompisar[], int n)
{
    // SKRIV FUNKTIONEN som presenterar vektorns alla namn på skärmen!
}

void avsluta(char *kompisar[], int n)
{
    // SKRIV FUNKTIONEN som frigör allt dynamiskt allokerat minne!
}
```

Testfall

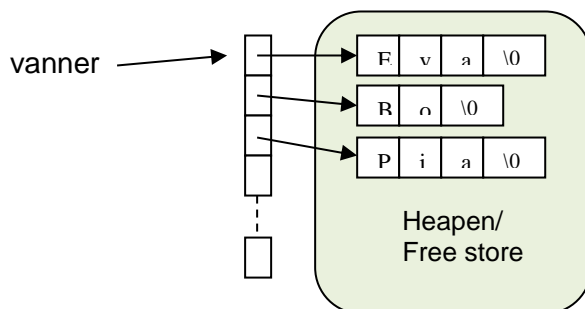
...

Tips

Använd den sorterings-algoritm som du finner under namnet "Enkelsort" på kurshemsidans **Steg 3** (i filen *Variabelbyte & sortering* under *Kompletterande material*). Denna sortering är mer lättförstådd än den sorteringsfunktion som presenteras i kursboken (vid namn *Bubblesort*). Tänk på att det nu är strängar i vektorn som ska sorteras, medan nämnda "Enkelsort" hanterar flyttal!

Med vektor menas en en-dimensionell array.

Vänner är en vektor (array) med pekare till char (strängar) som skapas dynamiskt.



Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.

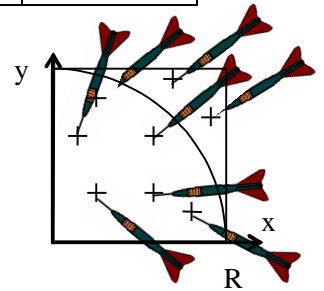


Kurskod 1DV433	Kurs Strukturerad programmering med C++			Institution Lnu/DFM	
Steg 5	Uppgift 11	Poäng 3	Utgåva 3	Datum 2012-10-01	Framtagen av Lö

Monte Carlo-metod för bestämning av Pi

Problem

Genom att slumpmässigt kasta pil på en kvadrat som är omsluten en kvartscirkel kan man experimentellt bestämma värdet på Pi med bra noggrannhet genom att anta att antalet pilar som träffar olika ytor står i proportion till dess areor. Med kvadratens sida = R får vi sambandet:



$$\frac{\text{Kvartscirkelearea}}{\text{Kvadratarea}} = \frac{\pi \cdot R^2 / 4}{R^2} = \frac{\pi}{4} \Rightarrow \frac{\text{Antal pilar under cirkelbågen}}{\text{Antal pilar i kvadraten}} = \frac{\pi}{4}$$

- Ett program för att undersöka denna metod att bestämma pi ska tas fram.
- Programmet ska använda ett antal funktioner som du ska få skriva.
- Programmet ska använda följande globalt åtkomliga definitioner:

```
const double R = 100;           // Cirkelns radie (=kvadratens sida) anges här

struct Coord                     // Datarepresentation av en punkt där pilen träffar
{
    double x, y;
};
```

- a) Skriv funktionen *kasta* som returnerar en koordinat av typen *Coord*. Kasten sker genom att dra slumpstal för x och y inom kvadratens område. Den slumpade koordinaten returneras i en struktur av typen *Coord*.
- b) Skriv den booleska funktionen *under* som tar en koordinat av typen *Coord* som argument och returnerar *true* om koordinaten ligger under cirkelbågen. Detta sker om $x^2 + y^2 < R^2$.
- c) Skriv funktionen *kastserie* som tar ett heltal, n, som argument. Funktionen ska kasta en pil n gånger och därefter skriva ut n-värdet, det med formeln ovan beräknade pi-värdet med 5 decimaler samt det relativa felet med 1 decimal. Funktionerna i a och b ska naturligtvis användas. Relativt fel beräknas med formeln: $\text{relFel} = \frac{\text{pi}_{\text{beräknat}} - \text{PI}}{\text{PI}} \cdot 100 (\%)$

Endast en rad ska skrivas ut vid anrop av denna funktion.

- d) Skriv slutligen ett program som beräknar pi på detta sätt med 100, 200, 300, ..., 10000 kast med pilen. Se testfall nedan (om du gjort rätt kommer du inte att få exakt samma värden som dessa).

Testfall

n	pi	Rel fel
100	3.12000	-0.7
200	3.20000	1.9
300	3.26667	4.0
400	3.15000	0.3
...
9800	3.13102	-0.3
9900	3.16646	0.8
10000	3.13360	-0.3

För det relativa felet behöver du ett "tabellvärde" för π .

Vissa kompilatorer har en konstant `M_PI` som anger ett noggrant värde på π .

Annars kan du skapa en konstant för π med `4*atan(1.0)`;



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/Dfm
Steg 5	Uppgift 12	Poäng 3-4	Utgåva 1	Datum 2012-12-18	Framtagen av Lö

Matrismultiplikation

Problem

Multiplikation av matriser används bland annat vid lösning av ekvationssystem och vid rotation av grafiska objekt i datorgrafik. Denna uppgift går ut på att du ska skriva ett program som multiplicerar 2 matriser med varandra.

Teori:

Vid multiplikationen $C = A \cdot B$ där A är en $m \cdot p$ -matris och B är en $p \cdot n$ -matris blir då C en $m \cdot n$ -matris. Det är ett villkor att antalet kolumner i A = antalet rader i B .

Elementen i C beräknas med formeln: $C(rad, kol) = \sum_{i=0}^{p-1} A(rad, i) \cdot B(i, kol)$

Exempel:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 3 & 4 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 \cdot 1 + 2 \cdot 3 + 3 \cdot 1 & 1 \cdot 1 + 2 \cdot 4 + 3 \cdot 2 \\ 4 \cdot 1 + 5 \cdot 3 + 6 \cdot 1 & 4 \cdot 1 + 5 \cdot 4 + 6 \cdot 2 \end{pmatrix} = \begin{pmatrix} 10 & 15 \\ 25 & 36 \end{pmatrix}$$

Uppgift

Skriv ett program som multiplicerar två inlästa matriser. Programmet ska som minst använda följande deklarationer:

```
const int MAXDIM = 5;
typedef double Matrix[MAXDIM][MAXDIM];

void readMatrix(Matrix M, int row, int col);
// Läser in en matris med row rader och col kolumner

void multMatrix(const Matrix A, const Matrix B, Matrix C, int m, int n, int p);
// Multiplicerar m•p-matris A med p•n matris B och sparar resultatet i m•n-matrisen C

void writeMatrix(const Matrix M, int row, int col);
// Skriver ut matrisen M
```

Globala variabler får inte användas!

Tips

Bry dig inte så mycket om exemplet när du ska skriva koden för multiplikationen. Beräkning av ett element i matrisen beskrivs bäst med formeln ovan. Den säger att ett element $C(rad, kol)$ erhålles om $A(rad, i) \cdot B(i, kol)$ summeras då i går från 0 till $p-1$. Summeringen läggs i en dubbel for-sats för att beräkna alla element i C .

Testfall

Se nästa sida!



Kurskod 1DV433	Kurs Strukturerad programmering med C++			Kurs Lnu/Dfm	
Steg 5	Uppgift 12	Poäng 3-4	Utgåva 1	Datum 2012-12-18	Framtagen av Lö

Testfall

Testa ditt program med de två matriser som finns nedan.

```
Matrismultiplikation
=====
Ange dimension för matris A <rad x col> med mellanslag mellan: 3 2
Skriv in matris A i fritt format
1 2
2 1
1 2
Ange dimension för matris B <rad x col> med mellanslag mellan: 2 2
Skriv in matris B i fritt format
3 2
1 5
-----
SVAR
      5.0      12.0
      7.0       9.0
      5.0      12.0
En gång till <J/N>:
```

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.



Kurskod 1DV433	Kurs Strukturerad programmering med C++				Kurs Lnu/Dfm
Steg 5	Uppgift 13	Poäng 3-6	Utgåva 1	Datum 2012-12-18	Framtagen av Lö

Black Jack

Problem

Skriv ett program som spelar Black Jack!

Spelregler för Black Jack:

Deltagaren spelar mot dealern (banken), oberoende av övriga spelare. Målet är att dra kort som tillsammans ger poängsumman 21, eller så nära 21 som möjligt, utan att överskrida 21.

Ess räknas som 1 eller 11, klädda kort som 10, och övriga kort enligt sitt värde.

Är spelarens poängsumma närmare 21 än dealerns, så vinner han lika mycket som han satsat, utom när han har Black Jack, vilket är 21 på två kort. Då vinner spelaren 1,5 gånger sin insats. Om båda får Black Jack vinner ingen, utan insatsen återgår till spelaren (så kallad "stand off").

Har spelaren och dealern samma poäng och dessa är 17, 18, eller 19 vinner dealern. Om poängen är 20 eller 21 eller Black Jack vinner ingen utan insatsen återgår till spelaren (s.k. "stand off"). Black Jack slår 21.

Får spelaren en poängsumma som överstiger 21 förlorar han sin insats.

Det finns dessutom ett antal specialfall som vi inte tar hänsyn till i denna tillämpning.

Testfall

...

Tips

Implementera kortleken med en array med 52 kort som fylls med talen 1-52. Valören kan beräknas med värdet % 13 och färgen med värdet % 4. Kortleken ska kunna blandas.

Redovisning

Uppgiften ska redovisas enligt:

- Ett fungerande program som uppfyller ställda kriterier.
- Programkod, som är välskriven och lättläst, enligt den kodstandard som tillämpas under kursen.