

MasterMind Lab 4

head.h

```
#ifndef _MASTERM_
#define _MASTERM_

#include <iostream>
#include <cstdlib>
#include <cctype>
#include <string>
#include <ctime>
using namespace std;

//TAL_STORLEK får maximat vara 10
const int TAL_STORLEK = 4;

int mataIn();
int mnuGetPrint();
void SpelaMM();
void ShowHigh();
void PrintHelp();
void Avsluta();
void SlumpaTal(int a[]);

typedef int tal[TAL_STORLEK];

struct score
{
    char namn[20];
    int poang;
    time_t date;
};

#endif
```

Hemligt tal – slumpa.cpp

```
#include "head.h"

//Slumpar ett tal med lika många siffror som TAL_STORLEK
//Pre: TAL_STORLEK <= 10, vektor har TAL_STORLEK platser
//Post: Genererat ett slumptal med TAL_STORLEK antal siffror
void SlumpaTal(int a[])
{
    int temp[10] = {0,1,2,3,4,5,6,7,8,9};
    int i = 0, ant = 10, slump;
    while(TAL_STORLEK > i)
    {
        slump = rand() % ant;
        a[i] = temp[slump];
    }
}
```

```
        temp[slump] = temp[ant - 1];

        ant--;
        i++;

    }

}

int main()
{

    srand(time(0));

    int hemligt[TAL_STORLEK];

    SlumpaTal(hemligt);

    return 0;
}
```

Gissa tal – gissa.cpp

```
#include "head.h"

//*****
//isOnlyDigits: Funktionen kontrollerar om inskickad sträng består av
//siffror. Negativa tal godkänns.
//pre: true.
//post: returnerar true om strängen bara består av siffror, annars false.
//*****

bool isOnlyDigits(const string str)
{
    for(int i = 0; i < str.length(); i++){
        if(!isdigit(str[i]))
            return false;
    }
    return true;
}

//*****
//convertToInt: Funktionen konverterar en sträng bestående av siffror till
//ett heltal.
//pre: isOnlyDigits(stringToConvert) == true.
//post: returnerar heltalsvärdet av strängens siffror.
//*****

int convertToInt(char charToConvert)
{
    return charToConvert - '0';
}
```

```
//Tar emot gissat tal från användare
// Ser till inmatningen endast kan anta siffror samt är av rätt antal.
//Pre: TAL_STORLEK > 0, a[] har TAL_STORLEK platser
//Post: a[] har blivit tilldelat TAL_STORLEK antal siffror från användare.
void taEmotGissn(int a[])
{
    char gissn[100];
    bool AlltOK = true;

    gissn[TAL_STORLEK] = '\0';

    cin.getline(gissn, 100);

    do
    {
        //Kontrollerar så att det bara är tal inmatade
        while(!(isOnlyDigits(gissn)) || AlltOK == false)
        {
            cout << "Inte ett giltigt gissning, endast " << TAL_STORLEK
                << " siffror, försök igen: ";

            gissn[TAL_STORLEK] = '\0';

            cin.getline(gissn, 100);

            AlltOK = true;
        }

        AlltOK = true;

        //Sätter till ogiltigt värde
        for(int k = 0; k < TAL_STORLEK; k++)
            a[k] = -1;

        //För över siffror från inmatade värden
        for(int i = 0; i < TAL_STORLEK; i++)
        {
            a[i] = convertToInt(gissn[i]);
            if(a[i] < 0)
                AlltOK = false;
        }

        if(gissn[TAL_STORLEK] != '\0')
            AlltOK = false;
    }while((AlltOK == false) || (gissn[TAL_STORLEK] != '\0'));
}
```

```
int main()
{
    int Gissning[TAL_STORLEK];

    cout << "Gissa: ";

    taEmotGissn(Gissning);

    for(int i = 0; i < TAL_STORLEK; i++)
        cout << Gissning[i];

    return 0;
}
```

HighScore – highscore.cpp

```
#include "head.h"

//Lägger till person till highscorlistan.
//Lägger personen på rätt plats (stigande ordnat på poäng)
//Pre: scores[] har totplats + 1 antal platser, points >= 0, totplats >= 0.
//Post: Namn, Poäng och Tid har laggtts till i score på rätt plats.
void addScore(score scores[], const int points, int &totplats)
{
    int plats = totplats;
    char* tid;
    bool klart = false;

    for(int i = 0; i<totplats; i++)
        if((scores[i].poang > points) && (plats==totplats))
            plats = i;

    for(int i = totplats; i>=plats; i--)
        scores[i+1]=scores[i];

    cout << "Namn: ";
    cin >> scores[plats].namn;
    scores[plats].poang = points;
    scores[plats].date = time(NULL);

}

int main()
{
    score highscore[20];
    int p=0;
    int t=0;

    while(p!=-1)
    {
        cout << "Poang: ";
        cin >> p;
        if(p!=-1)
            addScore(highscore, p, t);
    }
}
```

```
        t++;  
    }  
  
    cout << " ** HIGHSCORE **" << endl  
        << "=====" << endl;  
  
    for(int i=0; i<t-1; i++)  
        cout << "Namn: " << highscore[i].namn << endl  
            << "Poang: " << highscore[i].poang << endl  
            << "Tid: " << ctime(&highscore[i].date) << endl;  
  
    return 0;  
  
}
```