Institut für Informatik
im Fachbereich Mathematik und
Informatik, Physik, Geographie
Justus-Liebig-Universität Giessen

Übungen zum Programmierkurs
Sommersemester 2009
20. Juni 2009

Probeklausur

(9.1) [0 Punkte]

Was wird ausgegeben?

```
#include <iostream>
#include <ostream>
#include <curses.h>

using namespace std;

int test(int x)
{
    if (x == 0) return 1;
    int z = 0;
    for (int i=0; i<x; i++)
    {
        z = z + test(x-1);
    }
    return z;
}

int main()
{
    int z = 0;
    for (int i= 0; i < 10; i++)
    {
        if (i%3 == 1)
        {
            z= z + i;
        }
   }
   cout << z << "\n";
   cout << test(7);
   return 0;
}
```

1,4,7

"12"

"5040"

(9.2) [0 Punkte]

Schreiben Sie eine Funktion, die das kleineste Element einer verketteten Liste zurückgibt. Der Funktion wird nur der Anker der verketteten Liste übergeben.

```
struct TListenKnoten
{
int data;
TListenKnoten *next;
};
```

```
int minder( TListKnoten *anker){
  TListKnoten *akt=anker; int min=akt->data;
  while(anker-next!=0){
    akt=anker->next;
    if(akt->data<min)min=akt->data;
  }
  return min;
}
```

```
int vokale(string str){
  int zahl=0;
  for(int i;i<str.length;i++){
    if(str[i]=='A'||str[i]=='a'||...)zahl++;
  }
  return zahl
}
```

(9.3) [0 Punkte]

Schreiben Sie eine Funktion, die eine Zeichenkette übergeben bekommt und die Anzahl der Vokale der Zeichenekette zurückgibt.

(9.4) [0 Punkte]

Schreiben Sie ein Programm, dass 10 Integer Zahlen vom Benutzer einliest und in ein Array speichert. Danach soll das Programm alle geraden Zahlen aus dem Array ausgeben.

```
int main(){
  int c[10];
  for(int i=0;i<10;i++){
    cin>>c[i];
  }
  for(int i=0;i<10;i++){
     if(c[i]%2==0)cout<<c[i]<<"\n";
  }
}
```

(9.5) [0 Punkte]

Was wird ausgegeben?

```
#include <iostream>
#include <ostream>
#include <curses.h>
using namespace std;

class Tier
{
protected:
   string farbe; int groesse;
public:
   Tier()
   {
      cout << "Ich bin ein Tier\n";
      farbe = "blau"; groesse =50;
   }

   Tier(string Farbe)
   {
      cout << "Ich bin ein grosses Tier\n";
      farbe = Farbe; groesse =100;
   }

   void zeigeFarbe()
   {
        cout << farbe << "\n";
   }

   void virtual zeigeGroesse()
   {
        cout << groesse << "\n";
   }
};

class Hase : public Tier
{
public:
   Hase()
```

```
   {
      farbe = "braun"; groesse = 30;
      cout << "Ich bin ein Hase\n";
   }

   void zeigeFarbe()
   {
        cout << "Hase: " << farbe << "\n";
   }

   void zeigeGroesse()
   {
        cout << "Hase: " << groesse << "\n";
   }
};

class Katze : public Tier
{
public:
   Katze()
   {
      farbe = "grau"; groesse = 20;
      cout << "Ich bin eine Katze\n";
   }
   Katze(string Farbe, int Groesse) : Tier(Farbe)
   {
        farbe=Farbe; groesse = Groesse;
   }

   void zeigeFarbe()
   {
   cout << "Katze: " << farbe << "\n";
   }

   void zeigeGroesse()
   {
         cout << "Katze: " << groesse << "\n";
   }
};


int main()
{
   Tier *t[3];
   t[0]= new Tier;
   t[1]= new Hase;
   t[2]= new Katze("weiss", 60);
   for(int i = 0; i<3; i++)
   {
        t[i]->zeigeFarbe();
        t[i]->zeigeGroesse();
   }
   Hase h;
   h.zeigeFarbe();
```

Ich bin ein Tier

Ich bin ein Tier\n ich bin ein Hase

Ich bin ein grosses Tier\n

0:blau\n50

1:braun\nHase: 30

2:weiss\nKatze: 60

Ich bin ein Tier\n ich bin ein Hase

Hase: braun

```
    h.zeigeGroesse();
    Katze k;
    k.zeigeFarbe();
    k.zeigeGroesse();
    return 0;
}
```

Ich bin ein Tier\n ich bin eine Katze
Katze: grau
Katze: 20