

Programmeringsteknik med C och Matlab

Skrivtid: 9-13.

Hjälpmedel: Pennor.

Antal uppgifter: 7.

Instruktioner. OBS! Läs igenom instruktionerna noggrannt innan du börjar lösa uppgifterna.

- Börja med att skriva ditt namn och personnummer på första bladet. Skriv sedan ditt kodnummer (från försättsbladet) **på varje blad**.
- Skriv i mån av plats dina lösningar direkt i tentamen.
- Om du behöver mer plats, skriv dina lösningar på extrablad. Lös bara en uppgift på varje extrablad. Se till att ditt **kodnummer** och **uppgiftens nummer** är tydligt markerade på varje extrablad du lämnar in.
- Se till att den C-kod som ingår i dina lösningar är vettigt indenterad.
- **Skriv tydligt.** Om vi inte kan läsa dina lösningar kan vi inte ge poäng för dem.
- Observera att tentamen är tryckt dubbelsidig. Läs alltså båda sidorna av varje blad.

Betygsättning: Totalt går det maximalt att uppnå 44 poäng.

- För betyget **3** (godkänt) krävs **22 poäng**.
- För betyget **4** krävs **29 poäng**.
- För betyget **5** krävs **36 poäng**.

Kodnummer: _____

Uppgift 1

(5 Poäng)

Betrakta följande C-program:

```
#include <stdio.h>

int main(void){
    double x = 5.45;
    int y = -3;
    char c = 'w';
    double z = x + 4.55;
    int w = 4;

    printf("%f\n", x); /* Utskrift 1 */

    printf("%d\n", w % 3); /* Utskrift 2 */

    x -= y;

    printf("%lf\n", x); /* Utskrift 3 */

    printf("%.2f\n", z); /* Utskrift 4 */

    printf("%c %c %d\n", c, '=', w); /* Utskrift 5 */
    return 0;
}
```

Vad skriver programmet ut när det körs?

Utskrift 1:	5.450000
Utskrift 2:	1
Utskrift 3:	8.450000
Utskrift 4:	10.00
Utskrift 5:	w = 4

Σ:

Kodnummer: _____

Uppgift 2

(3 Poäng)

Volymen av en kon med cirkulär bas är lika med basens yta gånger höjden delat med 3. Basens yta är lika med basens radie i kvadrat gånger π . Alltså har vi

$$V = \frac{r^2 \cdot \pi \cdot h}{3},$$

där V är volymen, r är basens radie och h är konens höjd.

Skriv en C-funktion som givet konens höjd och basens radie som inparametrar returnerar motsvarande cylinders volym. Använd närmrevärdet 3.1416 för π . Din funktion ska ha följande prototyp:

```
double coneVolume(double radius, double height);
```

Lösningsförslag:

```
double coneVolume(double radius, double height){
    /* Base area is radius squared times pi */
    double baseArea = radius * radius * 3.1416;

    /* Volume is base area times height divided by 3 */
    return (baseArea * height) / 3;
}
```

Σ :

Kodnummer: _____

Uppgift 3

(7 Poäng)

a) Skriv en funktion med prototypen

```
int sumOfSquares(int source[], int n);
```

som returnerar summan av kvadraterna av de första n talen i arrayen `source`.

(3 poäng)

b) Skriv en `main`-funktion som

1. deklarerar en `int`-array av längd 21,
2. fyller arrayen med talen $-10, -9, \dots, -1, 0, 1, \dots, 9, 10$,
3. använder funktionen `sumOfSquares` för att räkna ut summan av kvadraterna av talen i arrayen, och
4. skriver ut resultatet (följt av en radbrytning).

(4 poäng)

Lösningsförslag:

```
int sumOfSquares(int source[], int n){
    int result = 0;
    int i;

    for(i = 0 ; i < n ; i++){
        result += source[i] * source[i];
    }
    return result;
}
```

```
int main(){
    int array[21];
    int i,result;

    for(i=0 ; i < 21 ; i++)
        array[i] = i-10;
    result = sumOfSquares(array, 21);
    printf("Sum of squares: %d\n",result);
    return 0;
}
```

Σ:

Kodnummer: _____

Kodnummer: _____

Uppgift 4

(7 Poäng)

Betrakta följande C-funktioner:

```
int rose(int lily, int tulip){
    if(lily <= tulip || lily > 3){
        return tulip / 2;
    }else if(tulip + lily >= 8){
        return tulip + 1;
    }else{
        return lily -1;
    }
}
```

```
int leprichaun(int fairy, int troll){
    int dragon = -troll;
    int giant = fairy;
    int i = fairy;
    while( i >= dragon){
        i--;
        giant += (fairy - 1);
    }
    return giant;
}
```

Vilket returvärde ger följande tre anrop?

rose(2,6);	3
rose(3,2);	2
leprichaun(4,-2);	13

Σ :

Kodnummer: _____

Uppgift 5

(7 Poäng)

Du har blivit anställd av Umeå universitet för att skriva program som hanterar data om universitetets olika kurser. För att komma igång måste du till att börja med definiera en lämplig datatyp för att hantera kursdata.

De uppgifter om en kurs som är intressanta är

- Kursens namn (max 300 tecken),
- Institutionen som ger kursen (max 100 tecken),
- Antal tillfällen då kursen hittills givits (ett heltal), och
- De år kursen hittills givits (en array med max 30 heltal).

a) Definiera en datatyp **courseData** i C som är lämplig för att representera kursdata.

Tips: Representera namnet och institutionen som arrayer av **char**.

b) Skriv en funktion som tar en **courseData** som indata och på ett prydligt sätt skriver ut den information om kursen som finns i datatypen. Din funktion ska ha följande prototyp:

```
void printCourseData(courseData course);
```

Resultatet av ett anrop till funktionen kan till exempel se ut så här:

```
Kursens namn: Programmeringsteknik med C och Matlab
```

```
Institution: Datavetenskap
```

```
Kursen har getts 3 gånger.
```

```
År då kursen har givits: 2007 2008 2009
```

Tips: Du kan anta att namnet och institutionen har lagrats som strängar (dvs null-terminerade **char**-arrayer).

Lösningsförslag:

```
typedef struct{
    char courseName[301];
    char institute[101];
    int noOfInstances;
    int years[30];
} courseData;

void printCourseData(courseData course){
    int i;
    printf("Kursens namn: %s\n", course.courseName);
    printf("Institution: %s\n", course.institute);
    printf("Kursen har getts %d gånger.\n",course.noOfInstances);
    printf("År då kursen har givits: ");
    for(i = 0 ; i < course.noOfInstances ; i++){
        printf("%d ", course.years[i]);
    }
    printf("\n");
}
```

Σ:

Kodnummer: _____

Uppgift 6

(7 Poäng)

a) Skriv en funktion

```
double ** allocDoubleMatrix(int rows, int columns);
```

som allokerar en tvådimensionell double-array (dvs. en array av arrayer av double) med rows rader och columns columner. (4 poäng)

b) Skriv en funktion

```
void freeDoubleMatrix(double ** matrix, int length);
```

som använder free för att frigöra det minne som allokerats för den tvådimensionella double-arrayen matrix. (Funktionen kan anta att matrisen har length rader.) (3 poäng)

Lösningsförslag:

```
double ** allocDoubleMatrix(int rows, int columns){
    double ** matrix = (double**)calloc(rows, sizeof(double*));
    int i;

    for(i=0 ; i<rows ; i++){
        matrix[i] = (double *)calloc(columns,sizeof(double));
    }
    return matrix;
}
```

```
void freeDoubleMatrix(double ** matrix, int rows){
    int i;
    for(i=0 ; i<rows ; i++)
        free(matrix[i]);
    free(matrix);
}
```

Σ:

Kodnummer: _____

Kodnummer: _____

Uppgift 7

(7 Poäng)

Ett heltal n kallas *perfekt* om det

1. är positivt och
2. är summan av alla sina delare (inklusive 1 men exklusive n självt).

Det minsta perfekta talet är 6, som är summan av sina mindre delare 1, 2 och 3.

Även $28 = 1 + 2 + 4 + 7 + 14$ är ett perfekt tal.

Skriv en funktion

```
int isPerfect(int n);
```

som returnerar 1 (sant) om n är ett perfekt tal och 0 (falskt) om n inte är ett perfekt tal.

Tips. Ett tal m delar ett tal n om $n \% m$ är lika med 0.

Lösningsförslag:

```
int isPerfect(int n){  
  
    int sum = 1;  
    int i;  
    if(n <= 1) return 0;  
  
    for(i = 2 ; i <= (n/2) ; i++){  
        if(n%i == 0) sum += i;  
    }  
    return sum == n;  
}
```

Σ :