Programmering C: Lösningar till tentamen 2007-01-13

Observera att detta är förslag på lösningar. Det kan finnas andra lösningar som också är korrekta, och det kan hända att en del av lösningarna är mer omfattande än vad som krävs för full poäng på uppgiften. En del av lösningarna är kanske inte fullständiga, utan hänvisar bara till var man kan läsa svaret.

Uppgift 1 (1 p)

```
x = 1 - (1 - x*x) / sqrt(1 - y*y);
```

Uppgift 2 (1 p)

a) 4

b) 2

c) 2.5

Uppgift 3 (1 p)

i = 2, j = 2

Uppgift 4 (1 p)

```
struct Planet {
    char namn[30 + 1];
    double vikt;
    int befolkning;
};
```

Uppgift 5 (1 p)

```
struct Planet jupiter = { "Jupiter", 2e27, 0 };
```

Uppgift 6 (2 p)

```
void visa_planet(struct Planet planeten) {
    printf("Namn: %s\n", planeten.namn);
    printf("Vikt: %f\n", planeten.vikt);
    printf("Befolkning: %d\n", planeten.befolkning);
}
```

Uppgift 7 (2 p)

Alternativ 1:

```
struct Planet las_planet(void) {
    struct Planet planeten;
    printf("Ge namn: ");
    scanf("%s", planeten.namn);
    printf("Ge vikt: ");
    scanf("%lf", &planeten.vikt);
    printf("Ge befolkning: ");
    scanf("%d", &planeten.befolkning);

    return planeten;
}
```

Alternativ 2:

```
void las_planet(struct Planet *p) {
    printf("Ge namn: ");
    scanf("%s", p->namn);
    printf("Ge vikt: ");
    scanf("%lf", &p->vikt);
    printf("Ge befolkning: ");
    scanf("%d", &p->befolkning);
}
```

Uppgift 8 (2 p)

```
int befolkad(struct Planet planeten) {
    return planeten.befolkning > 0;
}
```

Uppgift 9 (2 p)

Alternativ 1:

```
int main(void) {
    struct Planet p1, p2;

    p1 = las_planet();
    p2 = las_planet();

    visa_planet(p1);
    visa_planet(p2);

    return 0;
}
```

Alternativ 2:

```
int main(void) {
    struct Planet p1, p2;

    las_planet(&p1);
    las_planet(&p2);

    visa_planet(p1);
    visa_planet(p2);

    return 0;
}
```

Uppgift 10 (3 p)

Alternativ 1:

```
struct Planet las_befolkad_planet(void) {
    struct Planet planeten;

    do {
        printf("Ange en planet med befolkning!\n");
        planeten = las_planet();
    }
    while (!befolkad(planeten));

    return planeten;
}
```

Alternativ 2:

```
void las_befolkad_planet(struct Planet *p) {
    do {
        printf("Ange en planet med befolkning!\n");
        las_planet(p);
    }
    while (!befolkad(*p));
}
```

Uppgift 11 (6 p)

```
#include <stdio.h>

int main(void) {
    int tal;
    int antal_udda = 0;
    int antal_jamna = 0;

    printf("Ge ett heltal (avsluta med 0): ");
    scanf("%d", &tal);

    while (tal != 0) {
        if (tal % 2 == 0)
            ++antal_jamna;
        else
            ++antal_udda;

        printf("Ge ett heltal (avsluta med 0): ");
        scanf("%d", &tal);
    }

    printf("Antal jämna: %d\n", antal_jamna);
    printf("Antal udda: %d\n", antal_udda);

    return 0;
}
```

Uppgift 12 (6 p)

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define MAX_TAL 1000

int main(void) {
    double tal;
    double talen[MAX_TAL];
    int antal;
    double summan;
    int i;
    double medel;
    double narmaste;

    printf("Ge ett reellt tal (avsluta med negativt): ");
    scanf("%lf", &tal);

    while (tal >= 0) {
        if (antal == MAX_TAL) {
            printf("För många tal. Programmet avslutas.\n");
            return EXIT_FAILURE;
        }
        talen[antal++] = tal;

        printf("Ge ett reellt tal (avsluta med negativt): ");
        scanf("%lf", &tal);
    }

    if (antal == 0) {
        printf("Inga tal. Programmet avslutas.\n");
        return EXIT_FAILURE;
    }

    summan = 0;
    for (i = 0; i < antal; ++i)
        summan += talen[i];
    medel = summan / antal;

    narmaste = talen[0];
    for (i = 1; i < antal; ++i)
        if (fabs(talen[i] - medel) < fabs(narmaste - medel))
            narmaste = talen[i];

    printf("Medel: %f\n", medel);
    printf("Närmast: %f\n", narmaste);

    return EXIT_SUCCESS;
}
```

Uppgift 13 (6 p)

```
#include <stdlib.h>
#include <stdio.h>

/* ... */

int main(void) {
    struct Planet p;
    FILE *tsut;

    tsut = fopen("planeter.txt", "w");
    if (tsut == NULL) {
        printf("Kunde inte öppna filen 'planeter.txt'.\n");
        return EXIT_FAILURE;
    }

    printf("Ange planeter. Avsluta med en planet med vikten 0.\n");
    p = las_planet();
    while (p.vikt > 0) {
        fprintf(tsut, "%s %f %d\n", p.namn, p.vikt, p.befolkning);
        p = las_planet();
    }
    fclose(tsut);

    return EXIT_SUCCESS;
}
```

Uppgift 14 (6 p)

```
#include <stdlib.h>
#include <stdio.h>

/* ... */

int main(void) {
    struct Planet p;
    FILE *tsin;
    int antal = 0;
    int antal_befolkade = 0;
    struct Planet tyngsta_befolkade_planeten;

    tsin = fopen("planeter.txt", "r");
    if (tsin == NULL) {
        printf("Kunde inte öppna filen 'planeter.txt'.\n");
        return EXIT_FAILURE;
    }

    while (fscanf(tsin, "%s %lf %d", p.namn, &p.vikt, &p.befolkning) != EOF) {
        ++antal;
        if (befolkad(p)) {
            if (antal_befolkade == 0)
                tyngsta_befolkade_planeten = p;
            else if (p.vikt > tyngsta_befolkade_planeten.vikt)
                tyngsta_befolkade_planeten = p;
            ++antal_befolkade;
        }
    }
    fclose(tsin);

    printf("Antal planeter: %d\n", antal);
    printf("Antal befolkade planeter: %d\n", antal_befolkade);
    printf("Tyngsta befolkade planeten: %s\n", tyngsta_befolkade_planeten.namn);

    return EXIT_SUCCESS;
}
```

Programmet läser planeter tills filen tar slut, vilket märks på att **fscanf** returnerar värdet **EOF**.

Uppgift 15 (6 p)

a)

```
struct Rymdimperium {
    struct Planet centralplaneten;
    struct Planet underplaneter[MAX_PLANETER];
```

```
    int antal_underplaneter;
};
```

b)

```
struct Rymdimperium las_rymdimperium() {
    struct Rymdimperium r;
    int i;

    printf("Mata in rymdimperiets centralplanet.\n");
    r.centralplaneten = las_befolkad_planet();
    printf("Hur många underplaneter finns det? ");
    scanf("%d", &r.antal_underplaneter);
    for (i = 0; i < r.antal_underplaneter; ++i)
        r.underplaneter[i] = las_befolkad_planet();

    return r;
}
```

c)

```
int main(void) {
    struct Rymdimperium imperiet;
    imperiet = las_rymdimperium();

    return 0;
}
```

Thomas Padron-McCarthy (Thomas.Padron-McCarthy@tech.oru.se) 8 januari 2006