Uppgifter och Lösningar

1. Förklara kortfattat och förtydliga med ett exempel följande begrepp: (3p)
   (a) Funktionsdeklaration
   (b) Fält
   (c) Pekare

2. Studera programmet i Bilaga A. Vad skriver programmet ut på skärmen? (3p)

   Svar:

   ```
   1
   21
   321
   4321
   54321
   ```

3. Följande enum är deklarerad:

```
1 enum DAG { MON, TIS, ONS, TOR, FRE, LOR, SON };
```

   Skriv funktionen `imorgon` som returnerar den dag som kommer imorgon.

   Funktionsprototypen är:
   `enum DAG imorgon(enum DAG idag)` (3p)

   Svar:

```
1 enum DAG imorgon(enum DAG idag)
2 {
3     if(idag == SON) return MON;
4     return idag + 1;
5 }
```

4. Volymen av en kon beräknas med formeln:

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

   där r är konens bas-radie och h är konens höjd. Skriv en funktion som givet inparametrarna r och h beräknar konens volym. Antag att π har definierats med följande kod:

```
1 #define M_PI 3.14159265359
```

(3p)

Tips: Tänk på datatyperna.

Svar:

```
double kon_volym(double r, double h)
{
    double v = M_PI * r * r * h / 3;
    return v;
}
```

5. Skriv en funktion som kvarderar (dvs. multiplicerar med sig själv) varje element i ett fält. Funktionen skall returnera antalet element som förändrats; om ett element hade värdet 0 blir $0^2 = 0$ och har således inte förändrats.

   Funktionsdeklarationen är:
   `int my_square(int v[], int langd)` (4p)

   Ex:

```
int v[] = { 5, 5, 3, 0, 4, 0 };
int v_len = sizeof(v) / sizeof(int);
int resultat = my_square(v, v_len);
```

   skulle resultera i värdena $\{25, 25, 9, 0, 16, 0\}$ i fältet `v[]` och värdet 4 i variablen `resultat`.

   Svar:

```
int my_square(int v[], int langd)
{
    int uppdaterade = 0;
    for(int i = 0; i < langd; i++) {
        if(v[i] != 0) {
            v[i] = v[i] * v[i];
            uppdaterade++;
        }
    }
    return uppdaterade;
}
```

6. Skriv ett program som genererar slumptal och sparar dessa i en text-fil. Antalet slumptal som skall genereras och dess intervall (dvs. största och minsta slumptal) besvaras av användaren. (Felhantering för öppnande och skrivning till fil behöver inte tas omhand.) (6p)

Körexempel:

```
Minsta slumptal: 10
Storsta slumptal: 20
Antal slumptal: 5
Ange Filnamn: slumptal.txt
```

Innehållet i filen `slumptal.txt` efter att programmet körts:

```
18
18
18
13
10
```

Svar:

```
int main(int argc, char* argv[])
{
    int min, max, antal, slumptal;
    char filnamn[255];

    srand(time(NULL));

    printf("Minsta slumptal: ");
    scanf("%i", &min);
    printf("Storsta slumptal: ");
    scanf("%i", &max);
    printf("Antal slumptal: ");
    scanf("%i", &antal);
    printf("Ange Filnamn: ");
    scanf("%s", filnamn);

    FILE *fp = fopen(filnamn, "w");

    for(int i = 0; i < antal; i++) {
        slumptal = rand() % (max - min + 1) + min;
        fprintf(fp, "%i\n", slumptal);
    }

    fclose(fp);
    return 0;
}
```

7. Skriv en funktion som beräknar antalet bokstäver och siffror i en textsträng. Funktionen skall ta strängen som inparameter och returnera antalet bokstäver och antalet siffror via parametrar. ASCII-värdena för 'A'–'Z' är 65–90, 'a'–'z' är 97–122, och '0'–'9' är 48–57. Övriga tecken behöver inte tas hänsyn till. (6p)

   Ex:
   Följande text har 23 bokstäver och 1 siffror:

   ```
   En kort rad text med 1 siffra i.
   ```

   Svar:

```
1  void rakna(const char *str, int *bokstaver, int *tal)
2  {
3      *bokstaver = 0;
4      *tal = 0;
5
6      while(*str != '\0') {
7          if( (*str >= 'A' && *str <= 'Z') || (*str >= 'a' &&
               *str <= 'z') ) *bokstaver++;
8          else if (*str >= '0' && *str <= '9') *tal++;
9          str++;
10     }
11 }
```

8. Antag att vi skall bygga ett litet telefonregister med kortnummer som används för att ringa till kollegor inom ett företag. Följande post-typer är definierade:

```
1  #define MAX 255
2  #define NAMNMAX 32
3
4  struct Person {
5      char namn[NAMNMAX];
6      short kortnummer;
7  };
8
9  struct Register {
10     struct Person personer[MAX];
11     int antal;
12 };
```

   (a) Skriv en funktion som givet ett telefonregister och ett namn (sträng) returnerar kortnummret till den givna personen. Om namnet inte finns, returneras -1. (5p)

(b) Skriv en funktion som givet ett telefonregister och en person, lägger till denna i telefonregistret. Funktionen returnerar 0 om det fanns plats för personen, annars -1. (6p)

Tips: Det är tillåtet att använda standardfunktionen:
`int strcmp(char *s1, char *s2)`
Funktionen returnerar:

- ett negativt värde om `s1` kommer före `s2` (i lexikografisk ordning, d.v.s. i ett lexikon),
- ett positivt värde om `s1` kommer efter `s2`,
- 0 om strängarna är lika.

Svar:

```
short kortnummer(struct Register reg, char namn[NAMNMAX])
{
    for(int i = 0; i < reg.antal; i++) {
        if(strcmp(reg.personer[i].namn, namn) == 0)
            return reg.personer[i].kortnummer;
    }
    return -1;
}

int add(struct Register *reg, struct Person person)
{
    if(reg->antal == MAX) return -1;
    reg->personer[reg->antal] = person;
    reg->antal++;
    return 0;
}
```

9. Skriv en funktion, samt dess hjälpfunktion, som givet ett fält av tal, skriver ut ett histogram med freqvensen av varje tal. Histogrammet skall skrivas ut i *stigande* ordning. Ordningen på värdena i det givna fältet får förändras, därför tar vi hjälp av en sorterings-funktion som sorterar fältet innan vi skriver ut historgrammet.

   Studera följande programkod:

```
void histogram(int v[], int len);
void sortera(int v[], int len);
void skriv_pelare(int varde, int antal);

int main()
```

```
6  {
7    int v[] = { 5, 5, 3, 0, 4, 0 };
8    int v_len = sizeof(v) / sizeof(int);
9    histogram(v, v_len);
10   return 0;
11 }
12
13 void histogram(int v[], int v_len)
14 {
15   // Deklarationer
16   ...
17
18   // Sortera faltet
19   sortera(v, v_len);
20
21   // Implementera utskriften av histogrammet
22   ...
23 }
24
25 void sortera(int v[], int v_len)
26 {
27   // Implementationen av sortering av faltet
28   ...
29 }
30
31 void skriv_pelare(int varde, int freqvens)
32 {
33   // Skriver ut en pelare, dvs. freqvensen for det givna
       vardet.
34   printf("%i ", varde);
35   for(int i = 0; i < antal; i++) printf("X");
36   printf("\n");
37 }
```

En utskrift från programmet ser ut enligt följande:

```
0 XX
3 X
4 X
5 XX
```

Din uppgift är att implementera följande funktioner:

(a) Implementera `void sortera(int v[], int v_len)`. Funktionen skall sorte-

ra innehållet i det givna fältet (i stigande ordning). (6p)

(b) Fortsätt implementera `void histogram(int v[], int v_len)`. Funktionen skall skriva ut histogrammet i stigande ordningen (med hjälp av de andra hjälpfunktionerna). (5p)

Notera: Varje deluppgift kan lösas var för sig (även om den andra deluppgift inte löses).

Svar:

```
1  void histogram(int v[], int len)
2  {
3      // Deklarationer
4      int varde, antal;
5
6      // Sortera faltet
7      sortera(v, len);
8
9      // Implementera utskriften av ett histogram
10     varde = v[0];
11     antal = 1;
12
13     int i = 1;
14     while(i < len) {
15         if(v[i] == varde) {
16             antal++;
17         } else {
18             skriv_pelare(varde, antal);
19             varde = v[i];
20             antal = 1;
21         }
22         i++;
23     }
24     skriv_pelare(varde, antal);
25 }
26
27 void sortera(int v[], int len)
28 {
29     // Impelementationen av sortering av faltet
30     int tmp;
31     for(int i = 0; i < len - 1; i++) {
32         for(int j = i + 1; j < len; j++) {
33             if(v[j] < v[i]) {
34                 tmp = v[j];
35                 v[j] = v[i];
```

```
36                          v[i] = tmp;
37                      }
38                  }
39              }
40          }
```

A Programkod till Uppgift 2

```
1 #include <stdio.h>
2
3 #define MAX 5
4
5 int main()
6 {
7     int j;
8     for(int i = 1; i <= MAX; i++) {
9         j = i;
10        while (j > 0) printf("%i", j--);
11        printf("\n");
12    }
13    return 0;
14 }
```