

Övning 2 – Intelprogrammering

I alla uppgifter förutsätts att den gängse anropskonventionen för Linuxsystem (gcc/gas) och x64-arkitektur används.

1. Skriv en subrutin, `max`, som bestämmer vilket av två heltal som är störst.

Argument 1: Första heltalet

Argument 2: Andra heltalet

Returvärde: Det största av de två talen.

Rutinen ska anropas från följande C-program.

```
#include <stdio.h>
int max(int, int);

int main()
{
    int x,y,res;
    printf("Mata in två heltal\n");
    scanf("%d", &x);
    scanf("%d", &y);
    res = max(x,y);
    printf("Talet %d var störst av dem\n", res);
    return 0;
}
```

2. Antag att vi har en sekvens av n st positiva tal (array). Skriv en subrutin, `maxNum`, som ger oss det största talet i sekvensen.

Argument 1: Adress till talsekvensen

Argument 2: Antalet tal i sekvensen

Returvärde: Det största av talen i sekvensen

Rutinen ska ingå i ett bibliotek som exempelvis ska kunna anropas från följande C-program:

```
#include <stdio.h>
int maxNum(int *, int);

int main()
{
    int vect[5];
    int res;

    printf("Mata in fem heltal\n");
    scanf("%d", &vect[0]);
    scanf("%d", &vect[1]);
    scanf("%d", &vect[2]);
    scanf("%d", &vect[3]);
    scanf("%d", &vect[4]);
    res = maxNum(vect,5);
    printf("Talet %d var störst av dem\n", res);
    return 0;
}
```

3. Vi har en teckensträng som startar med siffror och avslutas med något som inte är en siffra. Skriv en rutin, `readInt`, som går igenom strängen och returnerar ett heltal (det tal vars representation utgörs av siffrorna i strängen).

Argument: Adress till strängen

Returvärde: Utläst heltal

Rutinen ska kunna anropas från följande C-program:

```
#include <stdio.h>
int readInt(char *);

int main()
{
    char str [10];
    int res;
    printf("Mata in ett tal! Avsluta med #!\n");
    fgets(str, 12, stdin);
    res = readInt(str);
    printf("Talet är: %d \n", res);
    return 0;
}
```

4. Förfina rutinen från förra uppgiften, så att strängen även kan innehålla ett godtyckligt antal blanktecken (noll eller flera) framför talet samt att talet kan inledas med ett tecken '+' eller '-' före första siffran.

5. Vi vill kunna jämföra två strängar med varandra och se vilken som kommer först i normal bokstavsordning. Rutinen `compare` ska vara skriven för det engelska alfabetet (inga å, ä, ö, alltså). Rutinen behöver bara klara ord med små bokstäver (gemener).

Argument 1: Adressen till första strängen.

Argument 2: Adressen till andra strängen.

Argument 3: Antal tecken i första strängen.

Argument 4: Antal tecken i andra strängen.

Returvärde: 1 om första strängen kommer före andra, 0 om strängarna är lika, -1 om andra strängen kommer före första.

Rutinen ska kunna anropas från följande C-program.

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

int compare(char *, char *, int, int);

int main ()
{
    char str1[15];
    char str2[15];
    int length1, length2, ret;

    printf("Skriv in första strängen!\n");
    fgets(str1, 80, stdin);
    printf("Skriv in andra strängen!\n");
```

```
fgets(str2, 80, stdin);
length1 = strlen(str1);
length2 = strlen(str2);
ret = compare(str1, str2, length1, length2);
if(ret > 0)
{
    printf("Första strängen kommer före den andra!\n");
}
else if(ret < 0)
{
    printf("Andra strängen kommer före den första!\n ");
}
else
{
    printf("Strängarna är lika!\n");
}
return 0;
}
```