

Laboration 1

Implementera följande uppgifter i MIPS-assembler. För utskrifter ska man använda den tillhandahållna subrutinen `print`. För inmatningar, etc., använd systemanrop beskrivna i Tabell 1.

Uppgift 1

Skriv ett program som frågar användaren efter en bokstav i alfabetet mellan b och y och sedan skriver ut den bokstav som kommer före och efter. Exempel: Om input är g blir output f h.

Uppgift 2

Skriv ett program som frågar användaren efter ett heltal och sedan skriver ut om talet är udda eller jämnt. Om talet är udda skriv ut odd och om talet är jämnt skriv ut even. Exempel: Om input är 4 blir output even.

Uppgift 3

Skriv ett program som frågar användaren efter ett heltal för bruttointkomst (brutto) och sedan skriver ut nettointkomsten (netto) som beräknas enligt följande pseudokod (där `threshold = 204000`, `low = 30` och `high = 50`):

```
if brutto <= threshold
    netto = brutto - brutto * low / 100
else
    netto = brutto - threshold * low / 100 - (brutto - threshold) * high / 100
```

Exempel: Om input är 40000 blir output 28000 och om input är 800000 blir output 440800.

Uppgift 4

Skriv ett program som frågar användaren efter ett årtal och sedan skriver ut om året är ett skottår eller inte (yes/no). Ett år är ett skottår om det är jämnt delbart med 400 eller jämnt delbart med 4 men inte med 100. Exempel: Om input är 2000 blir output yes och om input är 1900 blir output no.

Uppgift 5

Skriv ett program som frågar användaren efter ett heltal (n) och sedan beräknar värdet $1 + 2 + 3 + \dots$ tills summan överstiger n . Därefter ska summan samt det senast adderade talet skrivas ut. Exempel: Om input är 100 blir output 105 14 och om input är 50 blir output 55 10.

Uppgift 6

Skriv ett program som frågar användaren efter heltal och sedan adderar det inmatade talet till en summa. Vid varje inmatning skrivs summan ut. Programmet avslutas när användaren matar in talet 0. Exempel: Om input är 1 2 3 4 5 0 blir output 1 3 6 10 15.

Uppgift 7

Skriv ett program som frågar användaren efter n stycken heltal och sedan skriver ut det största och det minsta av de angivna talen. Vektorer (arrayer) får inte användas för talen. n ska vara en global variabel med värde 10. Exempel: Om input är 1 2 3 4 5 6 7 8 9 10 blir output 10 1.

Uppgift 8

Översätt följande C-program, som utför linjärsökning av en array, till MIPS-assembler:

```
#include <stdio.h>

int vec[] = {5,7,1,9,2,4,8,3,6};

main() {
    int s, i;
    int found = 0;
    int pos = -1;

    printf("Enter the number you are searching for: ");
    scanf("%d", &s);

    for (i = 0; !found && i < 9; i++) {
        if (vec[i] == s) {
            found = 1;
            pos = i + 1;
        }
    }

    if (found)
        printf("The position of the number is %d.\n", pos);
    else
        printf("The number is missing.\n");
}
```

Uppgift 9

Gör om programmet i Uppgift 8, så att linjärsökningen görs i en subrutin som anropas i huvudprogrammet. Subrutinen ska ha tre parametrar (startadress för array, storlek på array och det sökta talet) och ett returvärde (positionen).

Service	System call code	Arguments	Result
print_int	\$v0 = 1	\$a0 = integer	
print_float	\$v0 = 2	\$f12 = float	
print_double	\$v0 = 3	\$f12 = double	
print_string	\$v0 = 4	\$a0 = string	
read_int	\$v0 = 5		integer (in \$v0)
read_float	\$v0 = 6		float (in \$f0)
read_double	\$v0 = 7		double (in \$f0)
read_string	\$v0 = 8	\$a0 = buffer, \$a1 = length	
sbrk	\$v0 = 9	\$a0 = amount	address (in \$v0)
exit	\$v0 = 10		
print_char	\$v0 = 11	\$a0 = char	
read_char	\$v0 = 12		char (in \$v0)
open	\$v0 = 13	\$a0 = filename (string), \$a1 = flags, \$a2 = mode	file descriptor (in \$a0)
read	\$v0 = 14	\$a0 = file descriptor, \$a1 = buffer, \$a2 = length	num chars read (in \$a0)
write	\$v0 = 15	\$a0 = file descriptor, \$a1 = buffer, \$a2 = length	num chars written (in \$a0)
close	\$v0 = 16	\$a0 = file descriptor	
exit2	\$v0 = 17	\$a0 = result	

Tabell 1: Systemanrop¹

¹Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, 2014