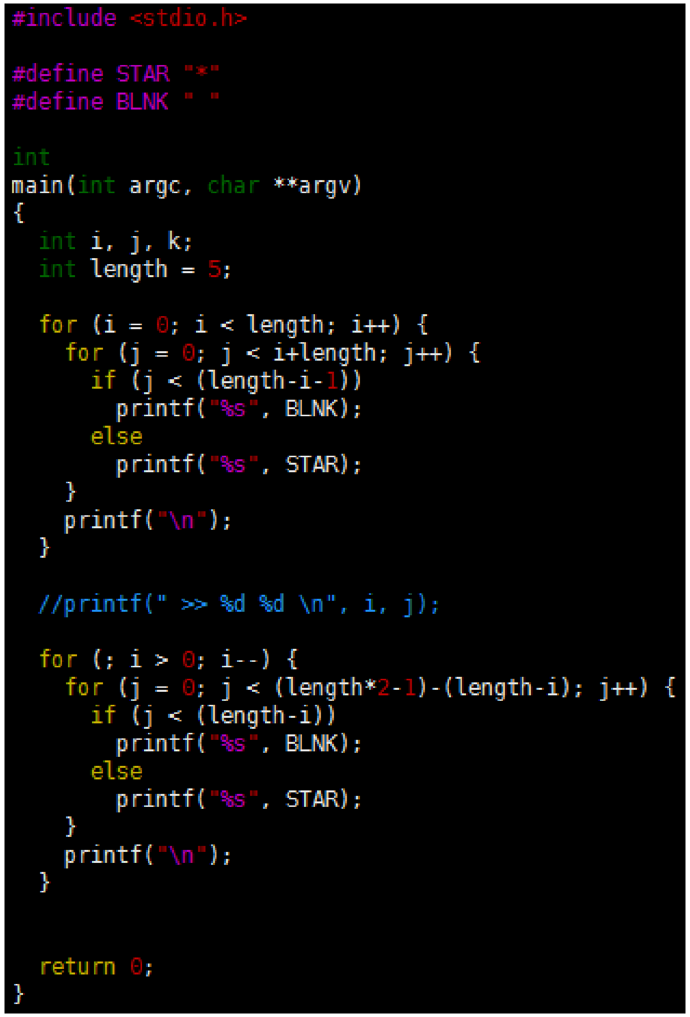
Computer Architecture – LAB 7

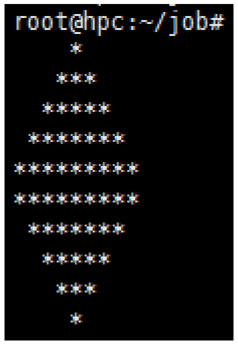
김원표
kwp@hallym.ac.kr

Lab 6

- Print Star using C



```c
#include <stdio.h>

#define STAR "*"
#define BLNK " "

int
main(int argc, char **argv)
{
  int i, j, k;
  int length = 5;

  for (i = 0; i < length; i++) {
    for (j = 0; j < i+length; j++) {
      if (j < (length-i-1))
        printf("%s", BLNK);
      else
        printf("%s", STAR);
    }
    printf("\n");
  }

  //printf(" >> %d %d \n", i, j);

  for (; i > 0; i--) {
    for (j = 0; j < (length*2-1)-(length-i); j++) {
      if (j < (length-i))
        printf("%s", BLNK);
      else
        printf("%s", STAR);
    }
    printf("\n");
  }


  return 0;
}
```



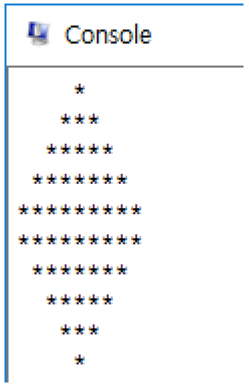
```
root@hpc:~/job#
    *
   ***
  *****
 *******
*********
*********
 *******
  *****
   ***
    *
```

Lab 6

Print Star using SPIM

```
.data
star: .asciiz "*"
space: .asciiz " "
newline: .asciiz "\n"
.text
.globl main
main:
    li $s0, 0   # for i
    li $s1, 0   # for j
    li $s5, 5   # for length

loop_j:
    add $s6, $s0, $s5            # length+i
    beq $s1, $s6, loop_i         # j == length+i ? loop_i: down

    sub $t1, $s5, $s0            # length-i
    addi $t1, $t1, -1            # -1

    blt $s1, $t1, print_space    # j < t1 ? print_space : down
    j print_star

loop_i:
    li $v0, 4                    # print line
    la $a0, newline
    syscall

    addi $s0, $s0, 1             # i++
    add $s1, $0, $0              # j = 0
    blt $s0, $s5, loop_j         # i < length ? loop_j: down

    j reverse_j

print_star:
    li $v0, 4                    # print star
    la $a0, star
    syscall

    addi $s1, $s1, 1             # j++
    j loop_j
```

```
print_space:
    li $v0, 4                      # print space
    la $a0, space
    syscall

    addi $s1, $s1, 1               # j++
    j loop_j

reverse_i:
    li $v0, 4
    la $a0, newline
    syscall

    addi $s0, $s0, -1              # i--
    add $s1, $0, $0                # j = 0
    bgt $s0, $0, reverse_j         # i > 0 ? reverse_j: down
    j exit

reverse_j:
    sll $t0, $s5, 1                # length * 2
    add $t0, $t0, -1               # -1
    sub $t1, $s5, $s0              # length - i
    sub $t2, $t0, $t1              # (length*2-1) - (length-i)

    beq $s1, $t2, reverse_i        # j == $t2 ? reverse_i: down
    blt $s1, $t1, print_space_r # j < length-i ? print_space : down
    j print_star_r

print_space_r:
    li $v0, 4
    la $a0, space
    syscall

    addi $s1, $s1, 1               # j++
    j reverse_j

print_star_r:
    li $v0, 4
    la $a0, star
    syscall

    addi $s1, $s1, 1               # j++
    j reverse_j

exit:
    li $v0, 10
    syscall
```

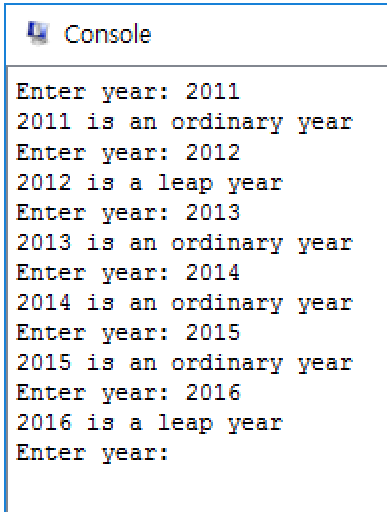


```
    *
   ***
  *****
 *******
*********
*********
 *******
  *****
   ***
    *
```

Lab 7

- **Leap Year and Ordinary Year**
  - year.asm

```
.data
year: .word 0
askyear: .asciiz "Enter year: "
ansleap: .asciiz " is a leap year\n"
ansordi: .asciiz " is an ordinary year\n"

.text
.globl main

main:
    li $v0, 4
    la $a0, askyear
    syscall

    li $v0, 5
    syscall
    sw $v0, year

    lw $t0, year
    li $t1, 4
    div $t0, $t1
    mfhi $t1
    bne $t1, $0, ordinary

    li $t1, 100
    div $t0, $t1
    mfhi $t1
    bne $t1, $0, leap

    li $t1, 400
    div $t0, $t1
    mfhi $t1
    bne $t1, $0, ordinary

leap:
    li $v0, 1
    lw $a0, year
    syscall

    li $v0, 4
    la $a0, ansleap
    syscall

    j finish

ordinary:
    li $v0, 1
    lw $a0, year
    syscall

    li $v0, 4
    la $a0, ansordi
    syscall

finish:
    j main
```



```
Enter year: 2011
2011 is an ordinary year
Enter year: 2012
2012 is a leap year
Enter year: 2013
2013 is an ordinary year
Enter year: 2014
2014 is an ordinary year
Enter year: 2015
2015 is an ordinary year
Enter year: 2016
2016 is a leap year
Enter year:
```

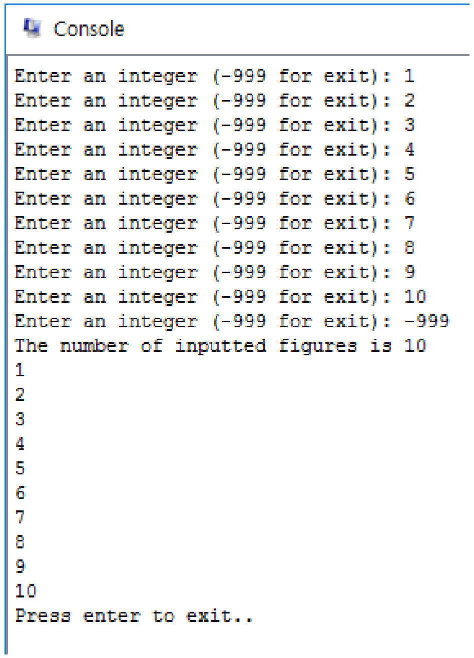
Lab 7

Implementing Arrays

- array.asm

```
.data
array: .word 0:10
ask: .asciiz "Enter an integer (-999 for exit): "
end: .word -999
resstr: .asciiz "The number of inputted figures is "
newline: .asciiz "\n"
bye: .asciiz "Press enter to exit.."

.text
.globl main

main:
    la $s0, array
    li $s1, 0

loop_1:
    li $v0, 4
    la $a0, ask
    syscall

    li $v0, 5
    syscall

    lw $t0, end
    beq $v0, $t0, exit

    sw $v0, 0($s0)

    addi $s0, $s0, 4

    addi $s1, $s1, 1
    j loop_1
```

```
exit:
    li $v0, 4
    la $a0, resstr
    syscall

    li $v0, 1
    add $a0, $0, $s1
    syscall

    li $v0, 4
    la $a0, newline
    syscall

    la $s0, array

loop_2:
    li $v0, 1
    lw $a0, 0($s0)
    syscall

    li $v0, 4
    la $a0, newline
    syscall

    add $s0, $s0, 4

    addi $s1, $s1, -1
    bne $s1, $0, loop_2

    li $v0, 4
    la $a0, bye
    syscall

    li $v0, 5
    syscall

    li $v0, 10
    syscall
```



```
Enter an integer (-999 for exit): 1
Enter an integer (-999 for exit): 2
Enter an integer (-999 for exit): 3
Enter an integer (-999 for exit): 4
Enter an integer (-999 for exit): 5
Enter an integer (-999 for exit): 6
Enter an integer (-999 for exit): 7
Enter an integer (-999 for exit): 8
Enter an integer (-999 for exit): 9
Enter an integer (-999 for exit): 10
Enter an integer (-999 for exit): -999
The number of inputted figures is 10
1
2
3
4
5
6
7
8
9
10
Press enter to exit..
```

Lab 7

- **Minimum and Maximum Numbers**
  - min_max.asm

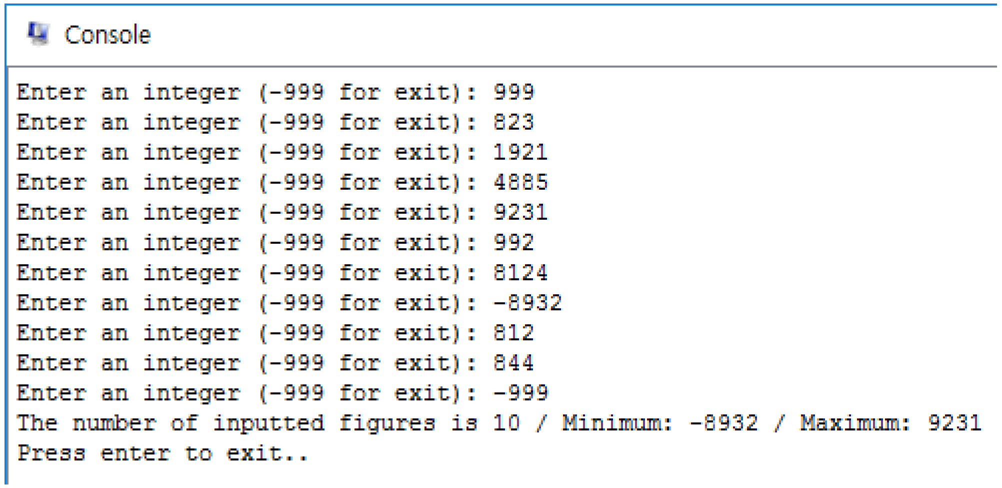
```
.data
array: .word 0:10
endmark: .word -999
ask: .asciiz "Enter an integer (-999 for exit): "
resstr: .asciiz "The number of inputted figures is
minstr: .asciiz " / Minimum: "
maxstr: .asciiz " / Maximum: "
newline: .asciiz "\n"
bye: .asciiz "Press enter to exit.."

.text
.globl main

main:
    la $s0, array
    li $s1, 0
loop_1:
    li $v0, 4
    la $a0, ask
    syscall

    li $v0, 5
    syscall

    lw $t0, endmark
    beq $v0, $t0, exit

    sw $v0, 0($s0)

    addi $s0, $s0, 4

    addi $s1, $s1, 1
    j loop_1
```

```
exit:
    li $v0, 4
    la $a0, resstr
    syscall

    li $v0, 1
    add $a0, $0, $s1
    syscall

    la $s0, array
    lw $t0, 0($s0)

    lw $t1, 0($s0)
loop_2:
    lw $t3, 0($s0)
    bge $t3, $t0, notmin

    add $t0, $0, $t3
notmin:
    ble $t3, $t1, notmax
    add $t1, $0, $t3
```

```
notmax:
    add $s0, $s0, 4
    addi $s1, $s1, -1
    bne $s1,  $0, loop_2

    li $v0, 4
    la $a0, minstr
    syscall

    li $v0, 1
    add $a0, $0, $t0
    syscall

    li $v0, 4
    la $a0, maxstr
    syscall

    li $v0, 1
    add $a0, $0, $t1
    syscall

    li $v0, 4
    la $a0, newline
    syscall

    li $v0, 4
    la $a0, bye
    syscall

    li $v0, 5
    syscall

    li $v0, 10
    syscall
```

Lab 7

- **Minimum and Maximum Numbers**
  - min_max.asm



```
Enter an integer (-999 for exit): 999
Enter an integer (-999 for exit): 823
Enter an integer (-999 for exit): 1921
Enter an integer (-999 for exit): 4885
Enter an integer (-999 for exit): 9231
Enter an integer (-999 for exit): 992
Enter an integer (-999 for exit): 8124
Enter an integer (-999 for exit): -8932
Enter an integer (-999 for exit): 812
Enter an integer (-999 for exit): 844
Enter an integer (-999 for exit): -999
The number of inputted figures is 10 / Minimum: -8932 / Maximum: 9231
Press enter to exit..
```

- **과제**
  - year.asm   array.asm   min_max.asm 에 라인별로 주석을 달아 워드문서로 정리하여 제출

- **파일명 ex) ca_07_학번_이름.docx**
  - **스마트 캠퍼스 과제란 제출 – 파일명 엄수**

- **제출기한**
  - 11월 23일 23:59까지

- **수업시간 내 완료시 조교의 확인을 받고 퇴실 가능, 미확인시 결석처리**

OSLAB Hallym University