VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY FACULTY OF COMPUTER SCIENCE AND ENGINEERING

****

COMPUTER ARCHITECTURE-CO2007

REPORT OF ASSIGNMENT

TOPIC

FOUR IN ROW

Instructor: Pham Quoc Cuong

Class:cc04

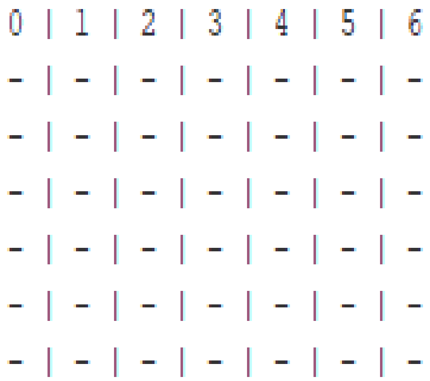
Name: Dinh Xuan Quang

ID:2053359

Table of content

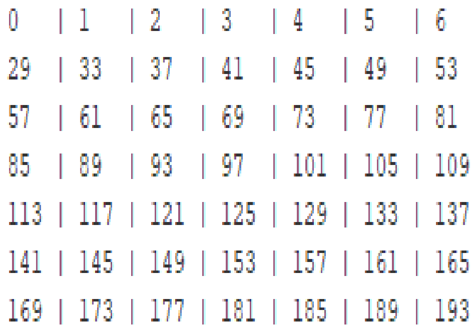
1. Draw the board

   Declare a string: .asciiz" 0 | 1 | 2 | 3 | 4 | 5 | 6 \n - | - | - | - | - | - | - \n - | - | - | - | - | - | - \n - | - | - | - | - | - | - \n - | - | - | - | - | - | - \n - | - | - | - | - | - | - \n - | - | - | - | - | - | - \n"



```
0 | 1 | 2 | 3 | 4 | 5 | 6
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | - | - | - | - | - | -
```

1.1This picture is table of connect 4 when I print in the terminal

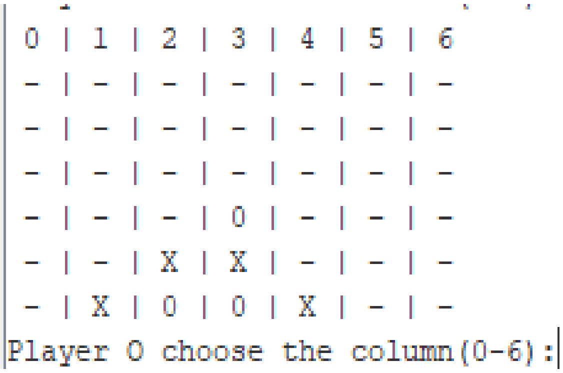


```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

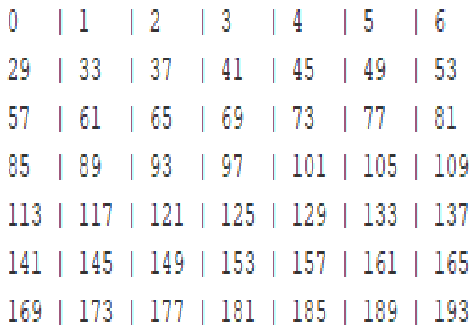
1.2 This picture contain the address corresponding picture 1.1

The value is the address that we use when we print X or O we will overwrite the value “-“ base on this address of the string

2. Print X or O and check the column is full



```
 0 | 1 | 2 | 3 | 4 | 5 | 6
 - | - | - | - | - | - | -
 - | - | - | - | - | - | -
 - | - | - | - | - | - | -
 - | - | - | O | - | - | -
 - | - | X | X | - | - | -
 - | X | O | O | X | - | -
Player O choose the column(0-6):
```



```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

_ Firstly the player will choose the column from 0 to 6

_ Then I will create the register A that contain value 169(the address of first column and first line , you can see on the right picture)

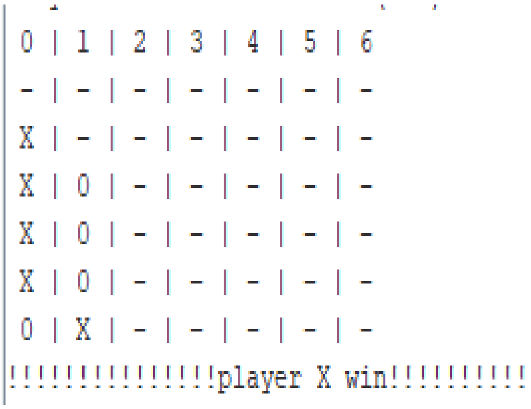
_ Then I use a loop to find the true column that player choose and I add the A=A+4

_ Next I will use loop and load byte the value at address A if value is equal to “-“ mean empty then I will assign the value X or O base on the player and get out of loop, when the value is not equal to “-“ I will subtract A=A-28(because the different of the address in the column is 28) to check the second value of the column and loop till when the value at address A is equal to “-“ or the value of A is less then 29 this mean the column is full .

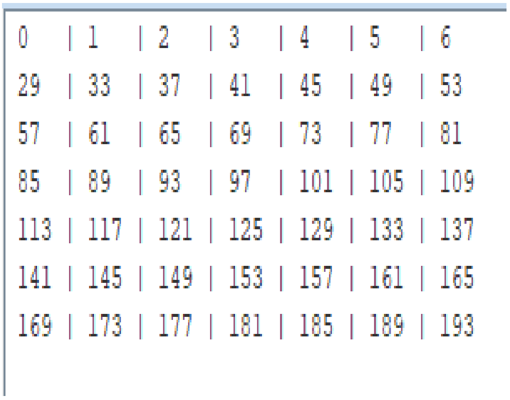
3. Check win

Check X and O win is similar so I will show the idea to check X in this part.

3.1 Check column



```
0 | 1 | 2 | 3 | 4 | 5 | 6
- | - | - | - | - | - | -
X | - | - | - | - | - | -
X | 0 | - | - | - | - | -
X | 0 | - | - | - | - | -
X | 0 | - | - | - | - | -
0 | X | - | - | - | - | -
!!!!!!!!!!!!!!!player X win!!!!!!!!!!
```



```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

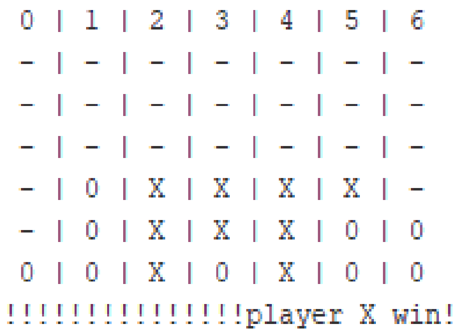
After player X turn I will check win X :

If the address of X that we just printed is greater 109 it mean that we don’t need to check column because the heigh is just 3

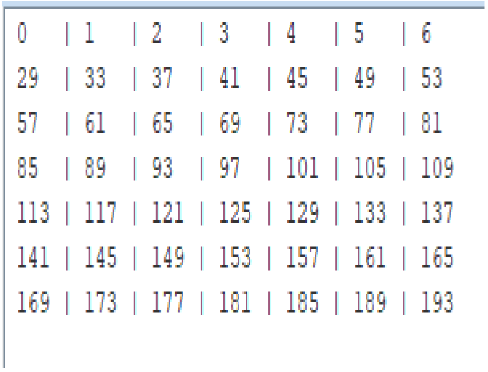
Otherwise, we will have a count check the value at under by adding A=A+28

Then we load byte at address A if it is equal X ,count=count+1 we continue to adding 28 to A and check until the count equal to 4 or it not equal to X we will move to checkline.

3.2Check line



```
0 | 1 | 2 | 3 | 4 | 5 | 6
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | - | - | - | - | - | -
- | 0 | X | X | X | X | -
- | 0 | X | X | X | 0 | 0
0 | 0 | X | 0 | X | 0 | 0
!!!!!!!!!!!!!!player X win!
```



```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

The address we have just print X is A, declare a new register B=A and count

Divide it into two part check left and check right

Using loop check left:

_ Firstly we check the column if the column is 0 we move to checkright

_ Then we subtract B=B-4 and load byte the value at B if it equal to X then we add count= count+1 and check if count=4 we will move to player X win , if count!=4 we jump to loop again.

Declare a new register B=A and using loop check right:

_ Firstly we check the column if the column is 6 we move to check diagonal.

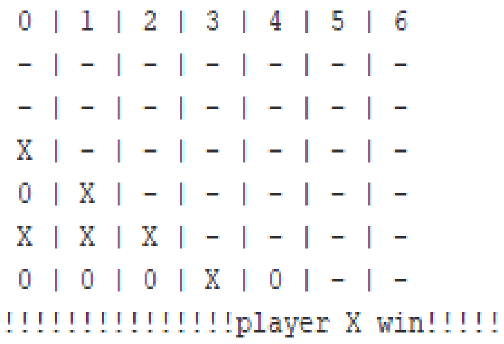
_Then we add B=B+4 and load byte the value at B if it equal to X then we add count= count+1 and check if count=4 we will move to player X win , if count!=4 we jump to loop again.

3.3 Check diagonally

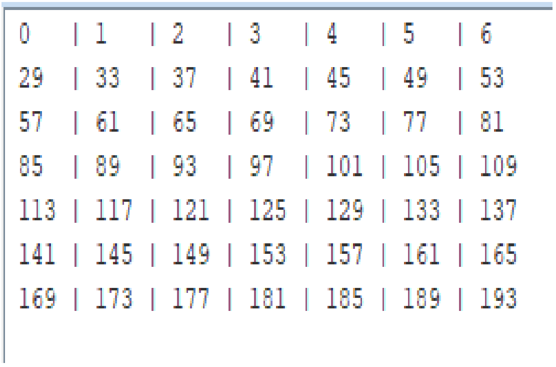
Divide two part:

- First part: Check Left UP and Right Down
    - Check Left Up
    - Check Right Down
- Second part: Check Left Down and Right Up
    - Check Left Down
    - Check Right Up

First part:



```
 0 | 1 | 2 | 3 | 4 | 5 | 6
 - | - | - | - | - | - | -
 - | - | - | - | - | - | -
 X | - | - | - | - | - | -
 0 | X | - | - | - | - | -
 X | X | X | - | - | - | -
 0 | 0 | 0 | X | 0 | - | -
!!!!!!!!!!!!!!!player X win!!!!!
```

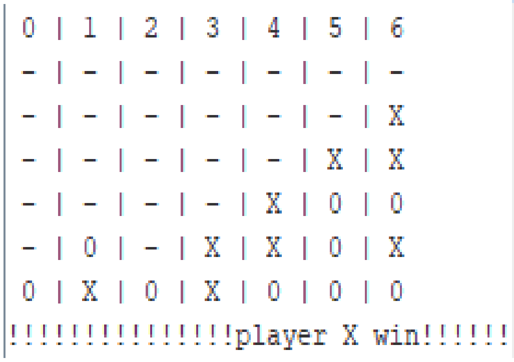


```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

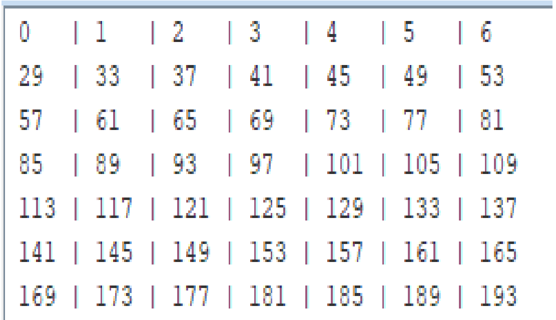
Declare a register B that contain address of X I have just printed ,C is a value of column, H is the height, count:

- Loop check Left Up:
    - Check if the height is equal to 6 then move to check Right Down
    - Check if the column is equal to 0 then move to check Right Down
    - Check and load byte the value at address B=B-32,if it is not equal to X move to check Right Down, if it equal to X, count =count+1 and check if count equal to 4 move to player X win, on the other hand,C=C-1 ,H=H+1 and go to loop check Left Up again.
- Loop check Right Down:
    - Check if the height is equal to 0 then move to the second part
    - Check if the column is equal to 6 then move to the second part
    - Check and load byte the value at address B=B+32,if it is not equal to X move to check Right Down, if it equal to X, count =count+1 and check if count equal to 4 move to player X win, on the other hand,C=C+1 ,H=H-1 and go to loop check Right Down again.

Second part:



```
 0 | 1 | 2 | 3 | 4 | 5 | 6
 - | - | - | - | - | - | -
 - | - | - | - | - | - | X
 - | - | - | - | - | X | X
 - | - | - | - | X | 0 | 0
 - | 0 | - | X | X | 0 | X
 0 | X | 0 | X | 0 | 0 | 0
!!!!!!!!!!!!!!player X win!!!!!!
```



```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

Declare a register B that contain address of X I have just printed ,C is a value of column, H is the height, count:

- Loop check Left Down:
    - Check if the height is equal to 0 then move to check Right Up
    - Check if the column is equal to 0 then move to check Right Up
    - Check and load byte the value at address B=B+24 if it is not equal to X move to check Right Down, if it equal to X, count =count+1 and check if count equal to 4 move to player X win, on the other hand,C=C-1 ,H=H-1 and go to loop check Left Down again.
- Loop check Right Up:
    - Check if the height is equal to 6 then move to the check draw
    - Check if the column is equal to 6 then move to the check draw
    - Check and load byte the value at address B=B-24,if it is not equal to X move to check Right Down, if it equal to X, count =count+1 and check if count equal to 4 move to player X win, on the other hand,C=C+1 ,H=H+1 and go to loop check Right Up again.

4.Check Draw

I have declare two register $a2,$a3 to count the number of X and O on the table.

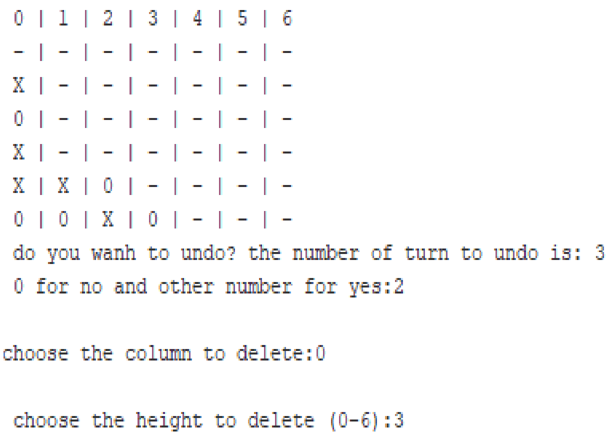
When print X addi $a2,$a2,1 and when undo X addi $a2,$a2,-1, O is similar

So when after check diagonally if a2+a3=42 then we will move to draw, if not we go to turn X or O again.

5.Undo

When we want to undo O:

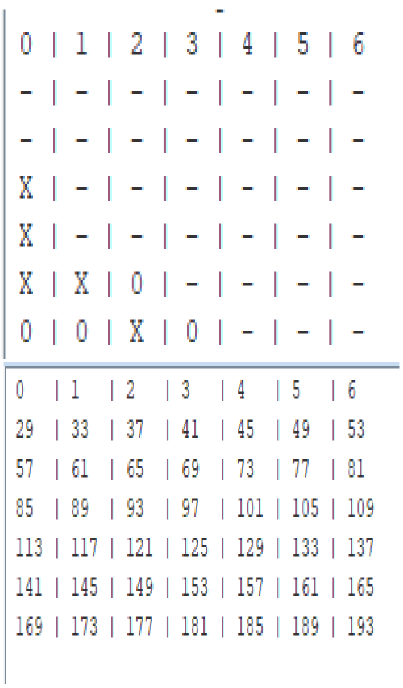
- Firstly it will check register $a3(the number of O on table), if it equal to 0 we can undo so move to print O, If not player can choose 0 for print X and other numbers for undo O.
- Then we will ask the player to choose the column(0-6) and the line(0-5) to delete
- Next we will check the position if it is not valid(out of range or not equal to O) then we print the string "your undo turn is not true please go to your turn again" will go back turn O.
- On the other hand when it valid we will find out the address A, the height and the column like we do in the print part



```
0 | 1 | 2 | 3 | 4 | 5 | 6
- | - | - | - | - | - | -
X | - | - | - | - | - | -
0 | - | - | - | - | - | -
X | - | - | - | - | - | -
X | X | 0 | - | - | - | -
0 | 0 | X | 0 | - | - | -
do you wanh to undo? the number of turn to undo is: 3
0 for no and other number for yes:2

choose the column to delete:0


 choose the height to delete (0-6):3
```



```
-
0 | 1 | 2 | 3 | 4 | 5 | 6
- | - | - | - | - | - | -
- | - | - | - | - | - | -
X | - | - | - | - | - | -
X | - | - | - | - | - | -
X | X | 0 | - | - | - | -
0 | 0 | X | 0 | - | - | -
```

```
0   | 1   | 2   | 3   | 4   | 5   | 6
29  | 33  | 37  | 41  | 45  | 49  | 53
57  | 61  | 65  | 69  | 73  | 77  | 81
85  | 89  | 93  | 97  | 101 | 105 | 109
113 | 117 | 121 | 125 | 129 | 133 | 137
141 | 145 | 149 | 153 | 157 | 161 | 165
169 | 173 | 177 | 181 | 185 | 189 | 193
```

Loop undo:

- Firstly we change the value at address C=A to "-"

- Next we check if the height is equal to 5 we end loop
- We check the value at address B=C-28 if it equal to “-“ then move to end loop, if not we load the value at address B and save byte the value at address A then
  C=C-28 and we go to loop again.
  After loop we will go to loop check win O and X:
- We will check the value at address A
  - If it equal to X go to check win X and go to loop check win O and X again
  - If it equal to O , then go to check win O and go to loop check win O and X again
  - On the other hand, exit the loop and go to turn X or turn O

5.Check error

I have declare two register that count the error of X and O, so when the count of error of player is equal to 3 , the other player will win

These are the cases that error in my program:

- Print in the column is full
- Undo the variable is not valid