Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 05

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

7. Write an assembly count-controlled loop program to display a row of 80 stars

Algorithms :

1.Start the program.
2. use Loop and print asterisks 80 times .
3.Stop the program.

Source Code :

```
.model small

.stack 100h

.data

mg1 db ?

mg2 db ?

.code

main proc


 mov ax,@data

 mov ds,ax


 mov cx,80

 jcxz skip

 top:

  mov ah,2

  mov dl,'*'

  int 21h

  loop top

  jmp skip


  skip:
```

```
mov ah,4ch
int 21h


main endp
end main
```

Output:

8. Write an assembly program to print the following series (for) 9 8 7 6 5 4 3 2 1

Algorithms :

1.Start the program.
2. Initialize ‘cx’ register with the value 9.

3.create a level named top, print 57,decrement the value of ‘dl’ register.Loop the level.

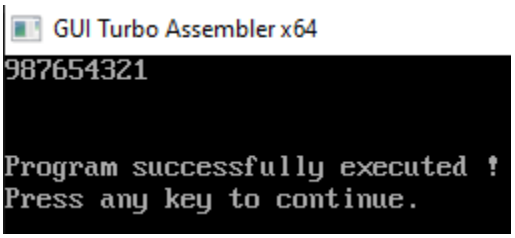
4.Stop the program.

Source Code :

```
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc
mov ax,@data
mov ds,ax
 mov cx,9
```

```
jcxz skip
top:
 mov ah,2
 mov bx,cx
 add bx,30h
 mov dl,' '
 mov dx,bx
 int 21h
 loop top
 jmp skip

 skip:
mov ah,4ch
int 21h

main endp
end main
```

Output:



9 . Write an assembly program to print the following series (for) 9 7 5 3 1

Algorithm:

1.Start the program.

2. Initialize ‘cx’ register with the value 5.

3.Create a level named top, print 57 ascii character .

4. Decrement the value of ‘dl’ register by 2. Loop the level.

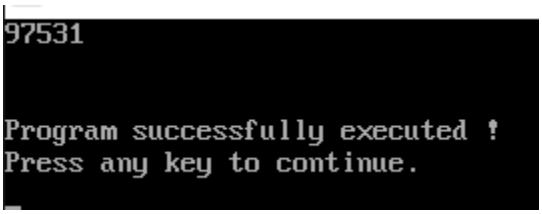
4.Stop the program.

Source Code :

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

 mov ax,@data
 mov ds,ax

 mov cx,10
 jcxz skip
 top:
  mov ah,2
  sub cx,1
  mov bx,cx
  add bx,30h
  mov dx,bx
  int 21h
  loop top
  jmp skip
```

```
skip:
mov ah,4ch
int 21h

main endp
end main
```

Output:



10. Write an assembly program to print the following series (for) 1 2 3 4 5 6 7 8 9

Algorithm:

1.Start the program.

2. Initialize ‘cx’ register with the value 9.

3.Create a level named top, print 49 ascii values character.

4. Increment the value of ‘dl’ register, Loop the level.

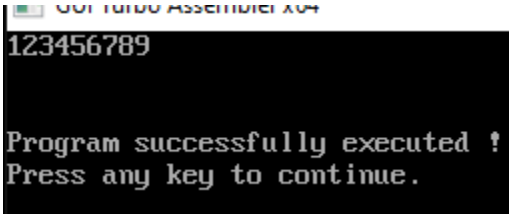
5. Stop the program.

Source Code :

```
.model small
.stack 100h

.code
main proc
```

```
mov cx,9
mov ah,2
mov dl,49
top:
int 21h

 inc dl
 loop top

exit:
mov ah,4ch
int 21h

main endp
end main
```

Output :



11. W rite an assembly program to print the following series (for) 8 6 4 2

Algorithm:

1.Start the program.

2. Initialize ‘cx’ register with the value 4.

3. create a level named top .

4. Print 56 ascii values , decrement the value of ‘dl’ register . Loop the level.

5. Stop the program.

Source Code :

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

 mov ax,@data
 mov ds,ax

 mov cx,8
 jcxz skip
 top:
  mov ah,2

  mov bx,cx
  add bx,30h
  mov dx,bx
  sub cx,1
  int 21h
  loop top
  jmp skip
```

skip:

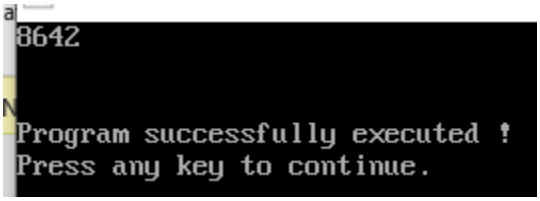
mov ah,4ch

int 21h

main endp

end main

Output :



12. Write an assembly program to print the following series (while) 9 8 7 6 5 4 3 2 1

Algorithm:

1.Start the program.

2. Initialize ‘dl’ register with the value 57.

3. create a level named while__, print 57,decrement the value of ‘dl’ register.

4. Compare the value of ‘dl’ register with the value 49.

5. If ‘dl’ register’s value is less then 49 then jump to exit level otherwise jump to while_ level.

6. Stop the program

Source Code:

.model small

.stack 100h

.data

mg1 db ?

mg2 db ?

```
.code
main proc

 mov ax,@data
 mov ds,ax

 mov ah,1
 int 21h
 sub al,30h
 mov bl,al

 MOV AH,2
 MOV DL,0AH
 INT 21H
 MOV DL,0DH
 INT 21H

 while_:
 cmp bl,0
 je exit
 mov ah,2
 mov cl,bl
 add cl,30h
 mov dl,cl
 int 21h
 dec bl
 jmp while_
```

```
exit:
mov ah,4ch
int 21h

main endp
end main
```

Output :