Report on

Arcade Game using Assembly Language (TASM)

Mini Project

TE Electronics & Telecommunication
Semester V

Submitted by

Samarth Shahu	**53**
Prithvi Sharma	**54**
Tanmay Lotankar	**70**

Under the guidance of

Miss. Manisha Joshi

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION
VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY
UNIVERSITY OF MUMBAI

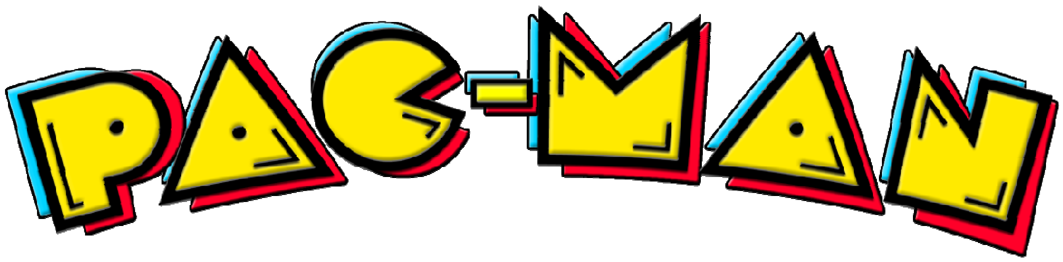
2019-2020

<u>**INTRODUCTION**</u>

The classic "*Pac-Man*" game was written in Assembly Language. It was one of the first games developed by this new department within Namco. The game was developed by a young 24-year-old employee over a year, beginning in April 1979, employing a nine-man team. It was based on the concept of eating, and the original Japanese title is "*Puckman*".

The video game franchise remains one of the highest-grossing and best-selling game series of all time, generating more than $14 billion in revenue (as of 2016) and $43 million in sales combined.

The game is regarded as one of the most influential video games of all time. The game was estimated to have had 30 million active players across the United States in 1982. In a 1983 interview, the developer said that though he did expect *Pac-Man* to be successful. It overtook "*Asteroids*" as the best-selling arcade game in North America, grossing over US$1 billion within a year, surpassing the highest-grossing film of the time, *Star-Wars*. Arcade machines retailed at around $2400 each and sales totaled around $1 billion (equivalent to $2.61 billion in 2018), within 18 months of release.



Our game “Piyu” was inspired by *Pac-Man*. However, we tried to remove complexity in the code while maintaining the gist of the game. The traditional Pac-Man uses various call back codes for calling of different pictures as Pac-Man, Monsters and Food items.

Instead we used ASCII value of different values to display Emojis instead of pictures to keep it simple and compact. By creating a whole X-Y plane of the Raster we brought movement in the main character by incrementation and decrementation of its 2D plotting variable values.

We placed sub-components of the games that are Hearts, Mines, B-Rolls and Borders.

Borders:- These are just the extreme values or restricted values of the X-Y plane counter. When the main character's location coincides with its location, the character stops and waits for external Keyboard interrupt for direction.

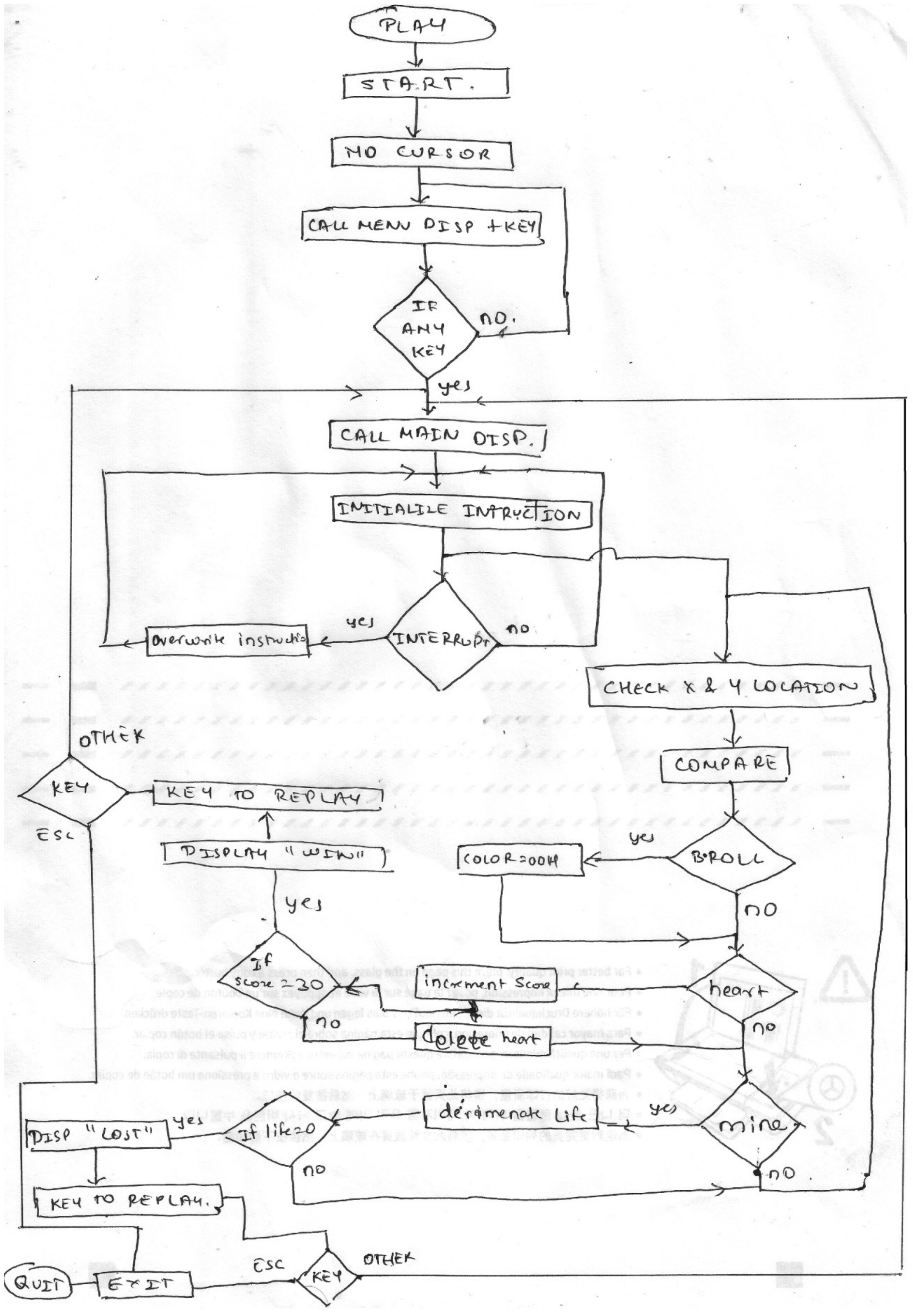
B-Roll:- This is just the background of the game which disappears when the position of character becomes equal to that of B-Roll.

Hearts:- By comparing position of the main character and heart we incremented the score counter and displayed it in the top left side of the main screen. As the position of character becomes equal to that of heart, one point is added to the score and the heart disappears. As the heart is deleted permanently from that position, hence no extra points are given if the character comes on the same location again. This way scoring of this game works.

Mines:- There are the end terminating operational locations which have the highest priority. These locations decrement the life by one. When the life drops down to zero the game ends.

Movement of Piyu:- It moves on the basis of X and Y incrementation and decrementation principles. Once any directional key is pressed i the keyboard, in goes on a infinite loop of execution until ay internal interrupt like mine, border or external interrupt like another directional key is pressed

WORKING FLOWCHART



REQUIREMENTS

- Personal Computer
- TASM software

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:-

- By using emoji instead of image on the main character, we reduced code length by 1/3.
- Since ASCII value was used instead of calling external functions, the game is fast, efficient and simple.
- The game is written in Assembly Language hence it is very efficient.
- One can understand its efficiency by observing that screenshot of the main screen utilizes 285Kb while the whole game with 5 different display screen is just made under 15Kb of data.
- Also by using Tlink and converting this into .exe form, the space required is only 3Kb.

DISADVANTAGES:-

- With the evolution of technology the software goes out of date.
- These games are rarely used.they cannot be played online.
- No multiplayer option available.
- Complex coding is required.
- Graphics, clarity and resolution used is not high definition.

FUTURE WORK AND CONCLUSION

FUTURE WORK:-

The arcade game industry is dying because of the upcoming age of the latest cutting age technology VR games where the player can experience the motion and game specific sensors. In many places the arcade games have nearly vanished. These games have a lack of diversity and innovation in the game design. Due to the decreasing revenues of the arcade games and the introduction of many high defined graphic games it seems to appear that arcade games have no future.

CONCLUSION:-

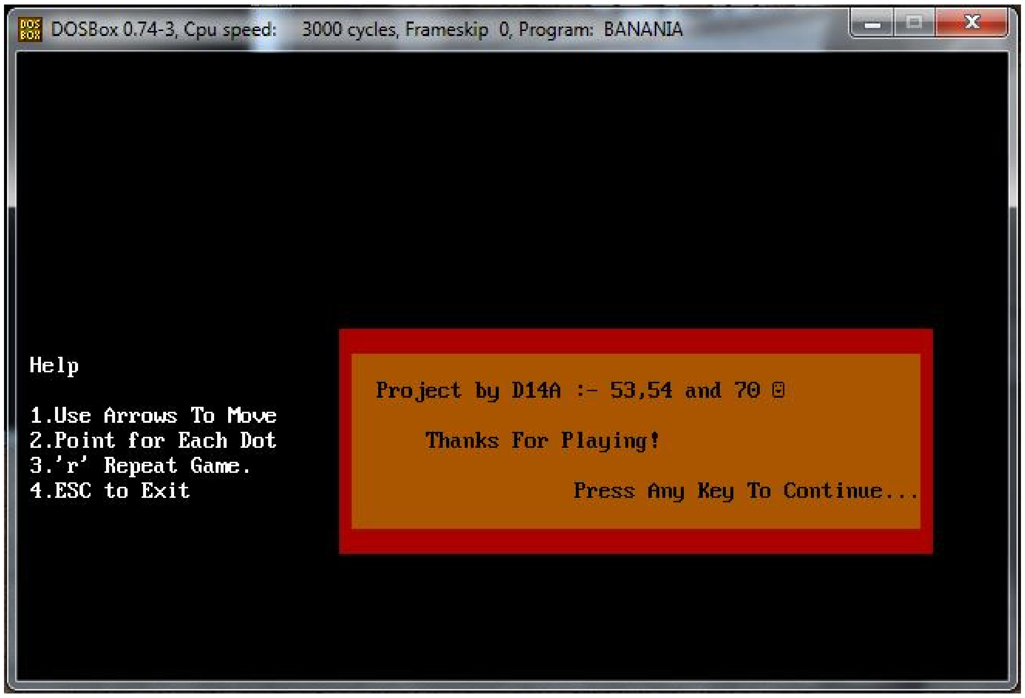
In this project we made an Aracde Game using 8086 software. An assembly language is one of the most efficient language. A good quality computer game needs around 3-4 GB of storage, in that much we can adjust 7 million games and still have enough amount of space to run live 8086 simulations in TASM simultaneously.But an arcade game requires very complex coding which can result in lot of consumption of time. Arcade games are very rarely used currently but these games are the roots of every modernised game present today.

REFERENCES

- https://www.udemy.com/x86-asm-foundations/
- https://www.udemy.com/course/programming-games-for-the-atari-2600/
- https://github.com/opferman/SixtyFourBits/commit/4672ff78f1b2a4e72efc40ce1ed6ac81ea23be11
- https://github.com/opferman/SixtyFourBits/commit/4672ff78f1b2a4e72efc40ce1ed6ac81ea23be11

APPENDIX

1. ENTRY SCREEN

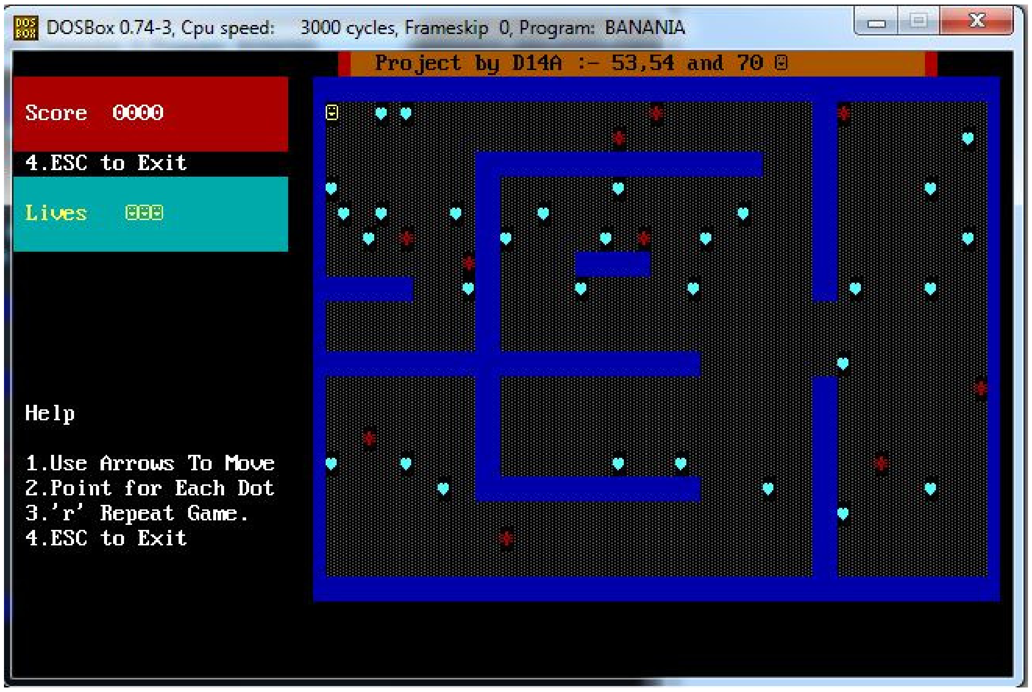


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: BANANIA

```
Help

1.Use Arrows To Move
2.Point for Each Dot
3.'r' Repeat Game.
4.ESC to Exit
```

```
Project by D14A :- 53,54 and 70 ☺

    Thanks For Playing!

              Press Any Key To Continue...
```

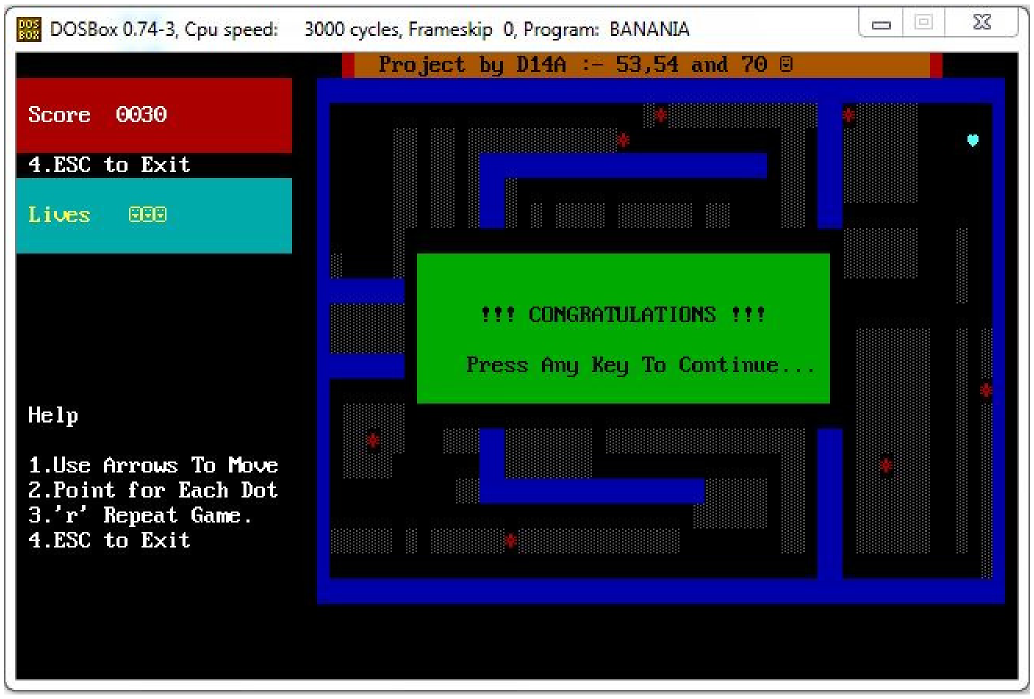
2. MAIN SCREEN



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: BANANIA

```
Project by D14A :- 53,54 and 70 ☺

Score  0000
4.ESC to Exit
Lives  ☺☺☺

Help

1.Use Arrows To Move
2.Point for Each Dot
3.'r' Repeat Game.
4.ESC to Exit
```

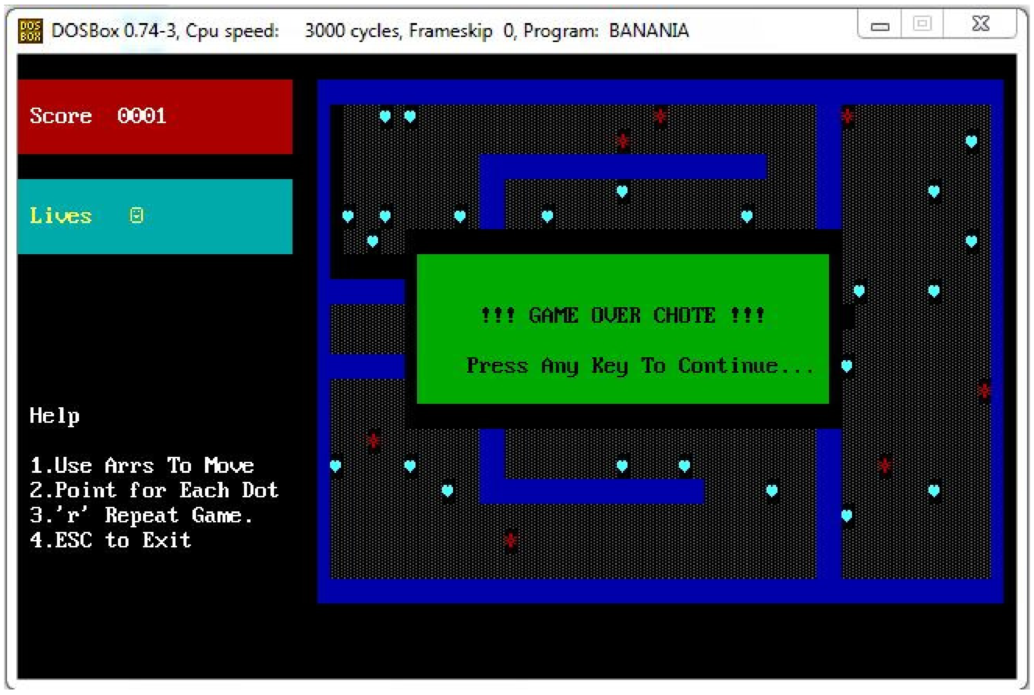
3. WINNING SCREEN



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: BANANIA

```
Project by D14A :- 53,54 and 70 ☺

Score  0030
4.ESC to Exit
Lives  ☺☺☺

!!! CONGRATULATIONS !!!

Press Any Key To Continue...

Help

1.Use Arrows To Move
2.Point for Each Dot
3.'r' Repeat Game.
4.ESC to Exit
```

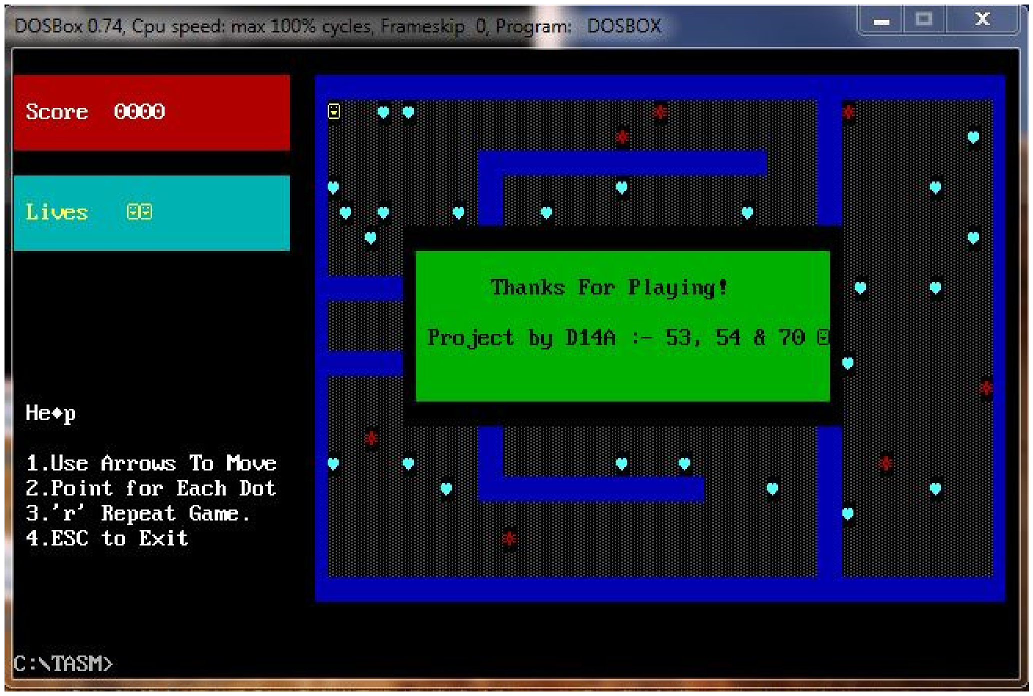
4. LOOSING SCREEN



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: BANANIA

```
Score  0001

Lives  ☺

!!! GAME OVER CHOTE !!!

Press Any Key To Continue...

Help

1.Use Arrs To Move
2.Point for Each Dot
3.'r' Repeat Game.
4.ESC to Exit
```

5. EXIT SCREEN


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: DOSBOX

```
Score  0000

Lives

Help

1.Use Arrows To Move
2.Point for Each Dot
3.'r' Repeat Game.
4.ESC to Exit

Thanks For Playing!

Project by D14A :- 53, 54 & 70

C:\TASM>
```