|  |  |
| --- | --- |
| **Ex No: 5** | **LOADER** |
| **Date of Exercise** | **3/02/2022** |

|  |
| --- |
| **AIM** |
| To simulate the concept of absolute loader. |
| **DESCRIPTION** |
| Absolute loader is a simple loader which performs loading process and need not to perform  linking and program relocation processes. |
| **ALGORITHM** |
| 1. Read the Header Record 2. Verify the name and length of the program. 3. Read first text record from the input file. 4. Process the following steps until the end record is reached. 5. If object code is character form then convert it to hexadecimal representation. 6. Move object codes to specified location in memory. 7. Read next text record from the input file one by one until reach the end record and 8. move them one by one to specified location. 9. If it reaches the end record close the files. |
| **SAMPLE INPUT & OUTPUT** |
| **Input.txt**  H^SAMPLE^001000^0035  T^001000^0C^001003^071009  T^002000^03^111111  E^001000  **Output**  enter program name: SAMPLE  name from obj. SAMPLE  001000 00  001001 10  001002 03  001003 07  001004 10  001005 09  002000 11  002001 11  002002 11 |
| **Code:** |

allLines = []

with open("/content/drive/MyDrive/Colab Notebooks/input\_ex5.txt") as inputText:

inputLines = inputText.readlines()

for line in inputLines:

allLines.append(line.split('^'))

headerRecord = allLines[0]

endRecord = allLines[len(allLines)-1]

programNameInput = input("Enter the name of the program: ")

programName = headerRecord[1]

print("Name from obj. "+programName)

if(programName==programNameInput):

for i in range(1,len(allLines)-1):

memoryStart = int(allLines[i][1])

# Joins all remaining strings after size in every text record

joinedString = ''.join(allLines[i][k] for k in range(3,len(allLines[i])))

j = 0

while(j+1<len(joinedString)):

print('00'+str(memoryStart)+' '+joinedString[j]+joinedString[j+1])

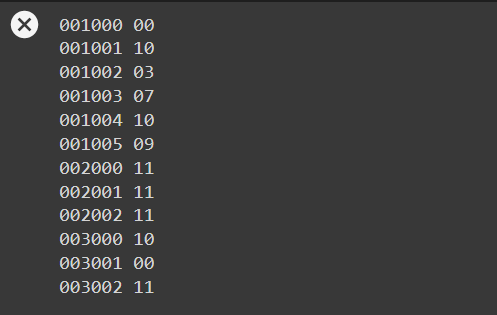
j+=2

memoryStart+=1

else:

print("Program names don't match. Execution ended")

**OUTPUT SCREENSHOT:**



**RESULT:**

Thus the program ran successfully and the output was verified.