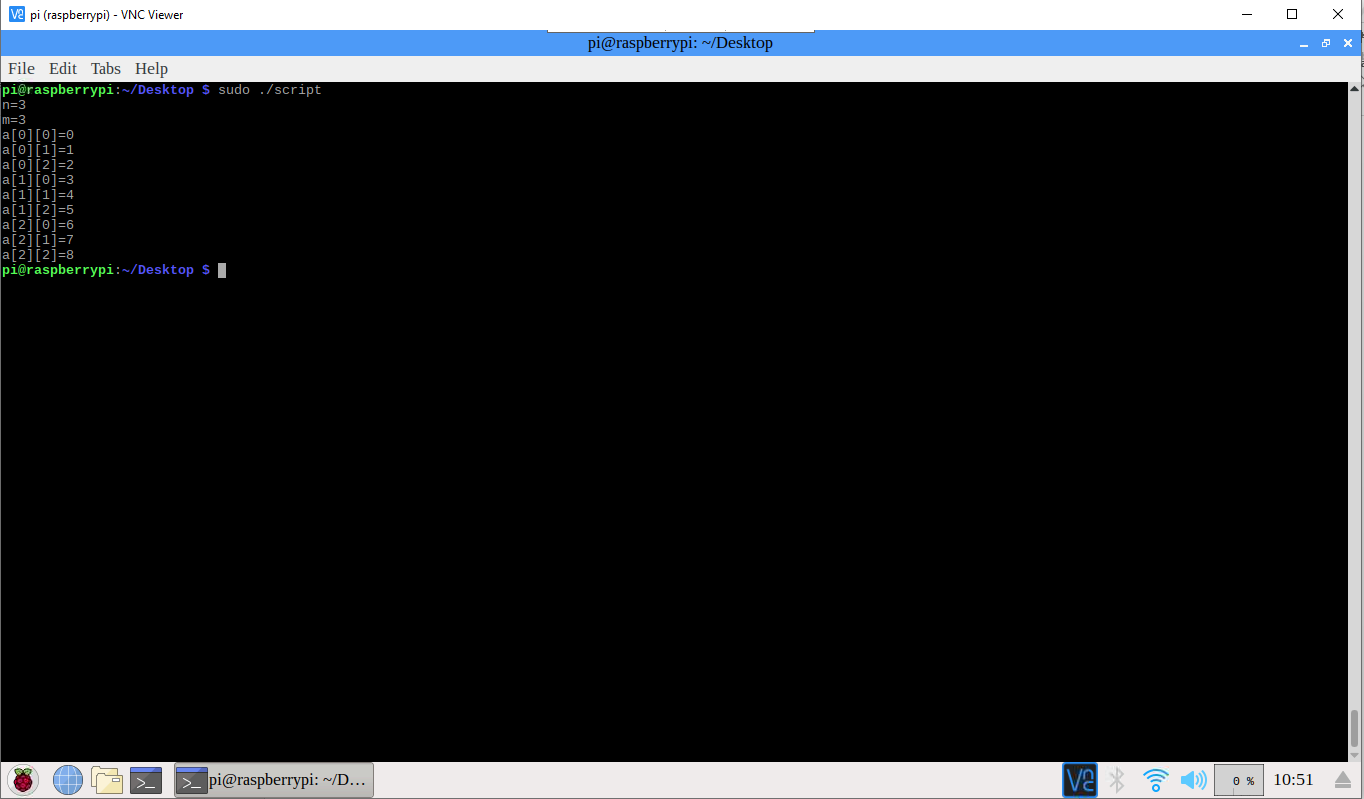
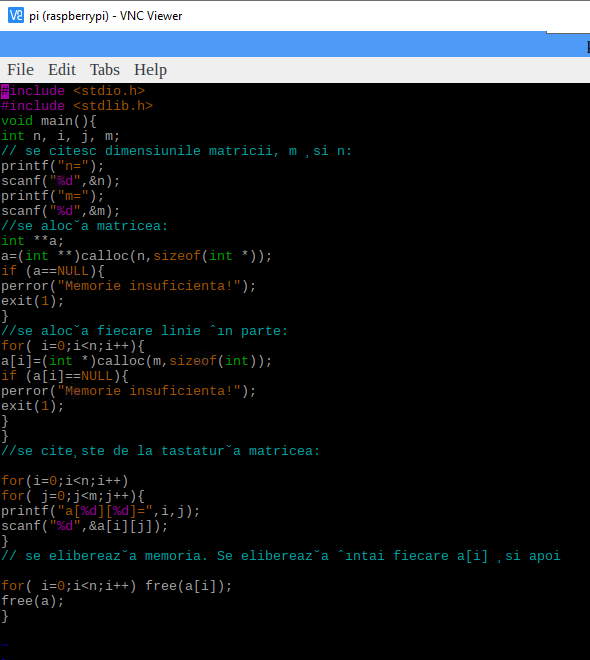
**Laborator 7 SO Cozmescu Daniel**

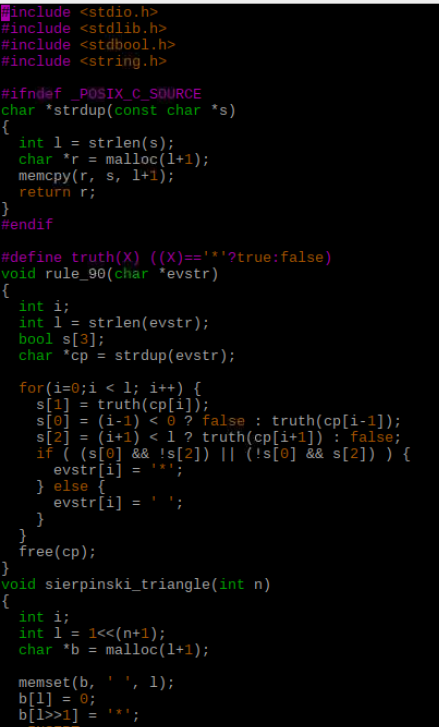
**Grupa 4LF781**

1. Se se ruleze exemplul cu matrice alocata dinamic din fisier.

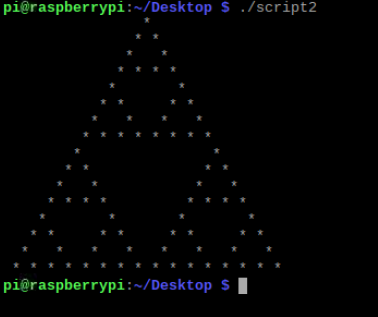




1. b) Sa se implementeze un fractal recursiv folosind matrice alocata dinamic.

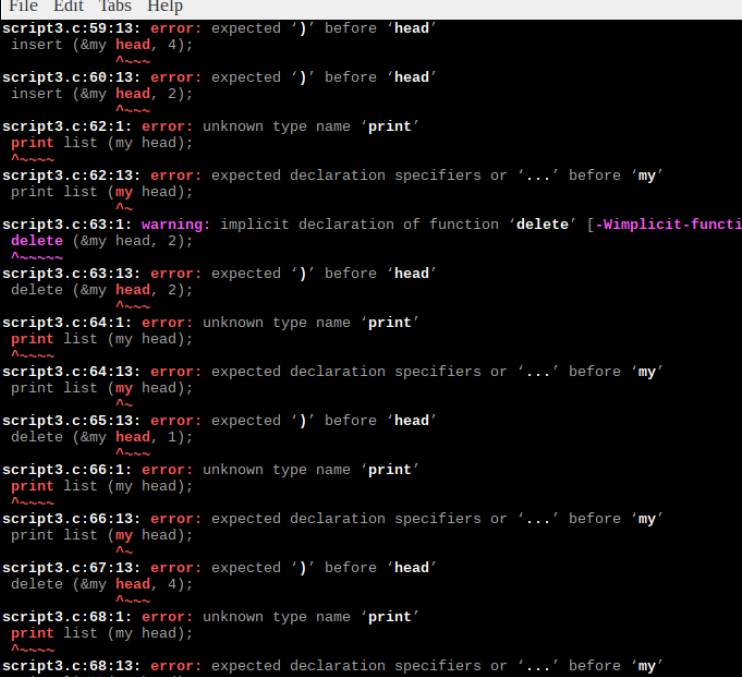




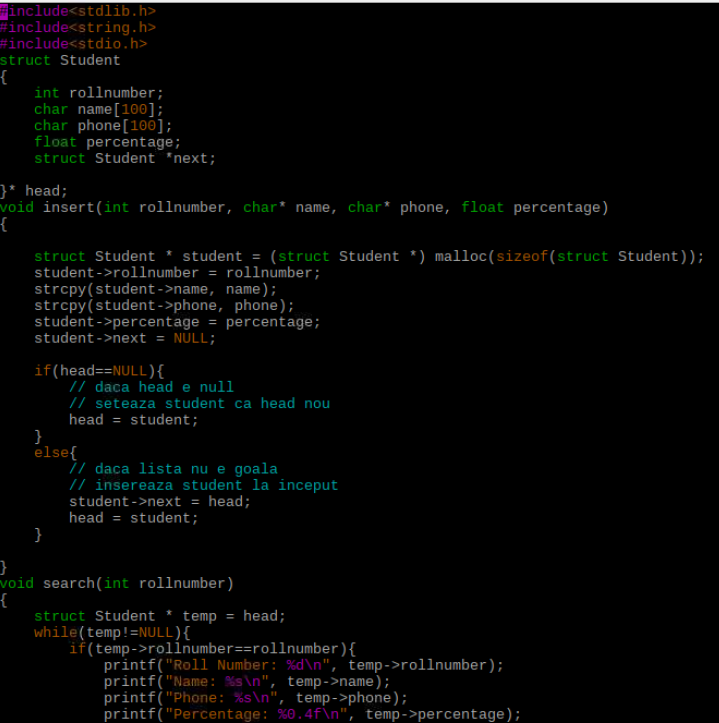


1. Se se ruleze exemplul de lista simplu inlatuita din fisier.

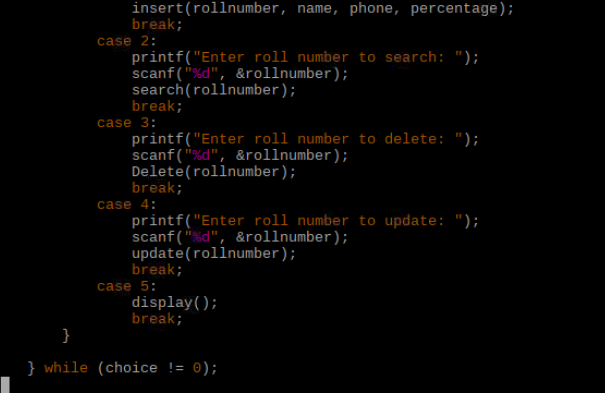
Nu am mai rulat deoarece codul din PDF are multe erori si nu am mai stat sa le corectez pe toate.

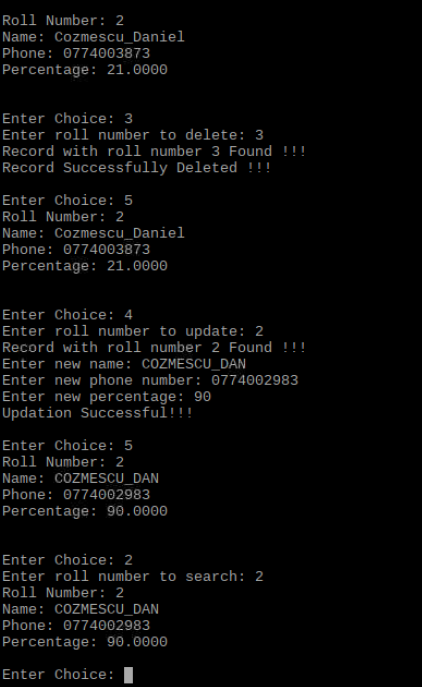
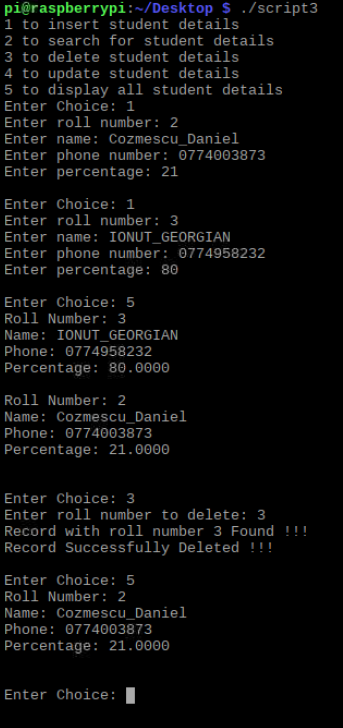


1. b) Sa se implementeze o structura student si operatiile de adaugare, modificare si stergere folosind liste simplu inlantuite.









Intr-un sistem de gestiunea a memoriei exista la un moment de timp dat urmatoarea lista inlantuita de blocuri libere 6KB, 12KB, 10KB, 22KB, 19KB, 7KB. In acest sistem apar urmatoarele cereri de memorie: 10KB, 14KB, 12KB, 8KB si 6KB. Folosind metodele de gestiunea a memorie a) Rotating-First-Fit b) Best-Fit raspundeti la urmatoarele intrebari:  
i) Ce blocuri de memorie libera vor fi asociate cererilor?

ii) Cum arata lista de spatii libere dupa prelucrarea tuturor cererilor?  
iii) Cati pasi de cautare se fac in lista de blocuri de memorie libere?

**BEST-FIT**

i)

|  |  |  |
| --- | --- | --- |
| Marime bloc memorie | Marime Job | Fragmentare Interna |
| 6KB | 6KB | - |
| 12KB | 10KB | 2KB |
| 10KB | 8KB | 2KB |
| 22KB | 14KB | 8KB |
| 19KB | 12KB | 7KB |
| 7KB | - | 7KB |
| TOTAL DISPONIBIL: 76KB | TOTAL FOLOSIT: 50KB | TOTAL: 26KB |

ii)

O zona de fragmentare lipseste deoarece marimea job-ului corespunde cu marimea blocului de memorie, iar o zona de Fragmentare Interna (cea de 7KB) este la valoare maxima in lipsa Jobului.

***Total marime bloc memorie = Total marime Job + Total Fragmentare Interna***

iii)

Cautare(10)=2, Cautare(14)=4, Cautare(12)=5, Cautare(8)=3, Cautare(6)=1 =>

Total pasi cautare = 15

**ROTATING-FIRST-FIT**

i)

|  |  |  |
| --- | --- | --- |
| Marime bloc memorie | Marime Job | Fragmentare Interna |
| 6KB | - | 6KB |
| 12KB | 10KB | 2KB |
| 10KB | - | 10KB |
| 22KB | 14KB | 8KB |
| 19KB | 12KB | 7KB |
| 7KB | 6KB | 1KB |
| TOTAL DISPONIBIL: 76KB | TOTAL FOLOSIT: 42KB | TOTAL: 34KB |

ii)

2 zone de joburi lipsesc deoarece prin algoritmul First-Fit se verifica daca pozitia actuala sau urmatoarea sunt disponibile si potrivite ca marime, iar in caz contrar se trece mai departe, deci exista 2 lipsuri in marime Job, insemnand ca din marimea totala de 5 joburi doar 4 au fost executate.

***Total marime bloc memorie = Total marime Job + Total Fragmentare Interna***

iii)

Cautare(10)=2, Cautare(14)=2, Cautare(12)=1, Cautare(8)=1, Cautare(6)=1 =>

Total pasi cautare = 7 pasi de cautare

1. Considerand cazul unui sistem de fisiere bazat pe inoduri cu dimensiunea blocului de 2 KB si stiind ca un inode contine 10 intrari pentru blocuri adresabile direct si ca fiecare referinta la un bloc de pe disc (numar al blocului) ocupa un cuvant de 32 de biti.  
   (a) Calculati dimensiunea maxima a unui fisier adresabil fara blocuri indirecte, cu blocuri simplu indirecte, cu blocuri dublu indirecte, respectiv cu blocuri triplu indirecte.  
   (b) Cate blocuri cu informatie de legatura sunt necesare in cazul unui fisier de 50MB?

(a)

2KB Marime bloc

10 intrari

32Bits/Adresa Bloc Disc

Blocurile directe stocheaza adresele blocurilor disc. Deci marimea fiecarui disc este 2KB = 211 Bytes.

Sunt 10 blocuri directe ce pot adresa: 10 x 2KB = 5\*211 bytes = 10240 bytes

Marimea fiecarui bloc este de 2KB = 211 bytes.

Marimea fiecarei adrese este de 32 biti. Deci fiecare bloc indirect poate adresa: 211 bytes / 4 bytes =29 blocuri

Deci,

1 bloc simplu indirect adreseaza 29 x 2KB = 29 x 211 = 220 Bytes

1 bloc dublu indirect adreseaza 29 x 29 x 211 = 229 Bytes

1 bloc triplu indirect adreseaza 29 x 29 x 29 x 211 = 238 Bytes

Deci

**Dimensiunea maxima a unui fisier este de 5\*211 + 220 + 229 +238 ≈ 275 Gigabytes**

Si

Fiecare bloc trebuie sa fie adresabil pentru a putea fi folosit, astfel avem 32 biti de adrese, deci putem adresa 232 ,fiecare bloc are 2KB=211 bytes deci putem adresa 232 x 211 = 243 bytes.

(b)

Numar blocuri necesare = Total dimensiune / 1 bloc dimensiune = 51,200 KB/2KB=25600 de blocuri

**Numar blocuri necesare = 25600**