

## Lucrarea de laborator nr 4

### Tema lucrării:

Analiza mecanismelor de planificare a proceselor sistemului de operare.

### Partea teoretică:

#### Planificare memoriei operative

Dacă procesele și spațiile libere sunt stocate într-o listă. Există mai mulți algoritmi pentru furnizarea memoriei unui proces.

1. Algoritmul primul interval potrivit. Funcționează după cum urmează, managerul de memorie parcurge lista de zone până când găsește o zonă liberă destul de mare. Această zonă este apoi împărțită în două părți: una este dat procesului, în timp ce celălalt rămâne neutilizat. Acest lucru se întâmplă întotdeauna cu excepția cazului statistic nerealist de corespondență exactă a zonei libere și proces. Acesta este un algoritm rapid, deoarece căutarea este redusă atât de mult pe cât posibil.

2. Algoritmul următorul interval potrivit funcționează în același mod ca primul algoritm, dar ori de câte ori găsește un fragment liber corespunzător, memorează adresa, unde sa oprit. Și data viitoare când algoritmul este apelat să caute, începe din același loc de unde s-a oprit data trecută în loc de a începe căutarea de la începutul listei.

3. Algoritmul cel mai potrivit interval caută și selectează cel mai apropiat fragment. În loc să împartă o zonă mare nealocată care ar putea fi necesară mai târziu, acest algoritm încearcă să găsească o parcelă care este aproape de dimensiunile reale necesare.

4. Algoritmul cea mai nepotrivit interval alege întotdeauna zona cea mai mare zonă liberă, din care rămâne o zonă după împărțire de dimensiuni suficiente și pot fi folosite în viitor.

Sunt date la realizare 5 procese cu volumul de memorie operativă necesar.

P1(15), P2(30), P3(5), P4(10), P5(20).

Lista zonelor de memorie liberă este

L(32, 40, 65, 26, 6, 18, 20, 32)

Care va fi algoritmul optimal pentru atribuirea memoriei.

Pentru primul potrivit

L(17, 40, 65, 26, 6, 18, 20, 32)

L(17, 10, 65, 26, 6, 18, 20, 32)

L(12, 10, 65, 26, 6, 18, 20, 32)

L(2, 10, 65, 26, 6, 18, 20, 32)

L(2, 10, 45, 26, 6, 18, 20, 32)

Lista finală L(10, 45, 26, 6, 18, 20, 32)

157

Pentru următorul potrivit

L(17, 40, 65, 26, 6, 18, 20, 32)

L(17, 10, 65, 26, 6, 18, 20, 32)

L(17, 10, 60, 26, 6, 18, 20, 32)

L(17, 10, 60, 16, 6, 18, 20, 32)

L(17, 10, 60, 16, 6, 18, 0, 32)

Lista finală L(17, 10, 60,16, 6, 18, 32)

159

Pentru cel mai potrivit

L(32, 40, 65, 26, 6, 3, 20, 32)

L(2,40, 65, 26, 6, 3, 20, 32 )

L(2,40, 65, 26, 1, 3, 20, 32())

L(2,40, 65, 26, 1, 3, 10, 32)

L(2,40, 65, 6, 1, 3, 10, 32)

Lista finală L(40, 65, 6,10, 32)

153

Pentru cel mai nepotrivit

L(32, 40, 50, 26, 6, 18, 20, 32)

L(32, 40, 20, 26, 6, 18, 20, 32)

L(32, 35, 20, 26, 6, 18, 20, 32)

L(32, 25, 20, 26, 6, 18, 20, 32)

L(12, 25, 20, 26, 6, 18, 20, 32)

Lista finală L(12, 25, 20, 26, 6, 18, 20, 32)

159

Optimal va fi algoritmul cel mai nepotrivit. Are acelaș timp utilizabil ca și algoritmul următorul potrivit, dar cu mai multe intervale libere utilizabile.

### **Algoritmi de înlocuire a paginii**

Când apare o eroare a paginii, sistemul de operare trebuie să selecteze o pagină de eliminat din memorie pentru a face loc paginii care urmează să fie mutată în memorie. Dacă pagina care este ștearsă a fost modificată în timp ce era în memorie, trebuie să fie scrisă pe disc pentru a actualiza copia stocată acolo. Totuși, dacă pagina nu a fost modificată (de exemplu, conține text de program), copia de pe disc este deja cea mai recentă și nu trebuie rescrisă. Apoi pagina de citit este pur și simplu citită în partea de sus a paginii care urmează să fie citită.

1. *Algoritmul FIFO* (First-In, First-Out). Sistemul de operare menține o listă cu toate paginile aflate în prezent în memorie, prima pagină fiind cea mai veche, iar paginile de la sfârșitul listei fiind cele mai recente. Când apare o eroare de pagină, pagina din capul listei este eliminată din memorie și o pagină nouă este adăugată la sfârșitul listei. Dacă se folosește algoritmul FIFO într-un magazin, poate îndepărta ceara de mustata, dar poate îndepărta și faina, sarea sau uleiul.

2. *Algoritmul a doua încercare* este algoritmul FIFO, care evită problema scoate paginile utilizate frecvent din memorie, bitul R al celei mai vechi pagini este examinat. Dacă este egal cu 0, pagina nu numai că rămâne mult timp în memorie, ci și nu este utilizată, deci este imediat înlocuită cu unul nou. Dacă bitul R este 1, atunci i se atribuie valoarea 0, pagina este mutată la sfârșitul listei și timpul de încărcare a acesteia este actualizat, adică pagina este considerată ca tocmai a intrat în memorie. Procedura continuă apoi.

3. *Algoritmul NRU* ((Not Recently Used) se șterge cea mai veche pagină folosind o căutare aleatorie în clasa nevidă cu numărul cel mai mic.

Când apare o eroare de pagină, sistemul de operare verifică toate paginile și le împarte în patru categorii pe baza valorilor curente ale biților R și M:

Clasa 0: fără accesări sau modificări.

Clasa 1: fără accesări, pagina schimbată.

Clasa 2: a fost accesată, pagina nu a fost schimbată.

Clasa 3: S-au produs accesarea, cât și schimbarea.

4. *Algoritmul LRU* (Utilizat cel mai puțin recent) - pagina care nu a fost folosită cel mai mult timp. Când apare o întrerupere de pagină, pagina care nu a fost folosită cel mai mult timp este descărcată din memorie. Această strategie de înlocuire a paginii este similară cu algoritmul NRU, dar pagina care a fost nefolosită cel mai mult timp este eliminată dar nu cea mai veche pagină.

SO are patru pagini. Timpul de încărcare, timpul ultimului apel și biții R și M pentru fiecare pagină sunt afișate mai jos.

Pagină	Încărcat	Ultimul apel	R	M
0	126	280	0	0
1	230	285	0	1
2	140	270	0	0
3	110	265	1	1

Ce pagină va fi eliminată din RAM atunci când se utilizează algoritmul:

1. Algoritmul FIFO

Va fi înlocuită pagina 3, deoarece este cea mai veche are timpul de încărcare 110.

2. Algoritmul a doua încercare.

Va fi înlocuită pagina 0, deoarece este cea mai veche cu bitul R=0. Paginile 3 este mai vechi, dar are bitul R=1

3. Algoritmul NRU

Va fi înlocuită pagina 0, deoarece este cea mai veche cu biții R=0 și M=0

4. Algoritmul LRU

5. Va fi înlocuită pagina 2, deoarece are cel mai mic timpul ultimului apel 270 și R=0 și M=0

### Organizarea memoriei paginii

Spațiul de adrese virtuale este împărțit în unități numite pagini. Unitățile corespunzătoare din memoria fizică sunt numite blocuri de pagini. Paginile și blocurile lor au întotdeauna aceeași dimensiune. În sistemele reale, sunt utilizate dimensiuni de pagină de 512 octeți.

Pe care pagină virtuală se va afla adresa fizică 20000 și care va fi deplasamentul?

Pentru pagina cu 4KB

0p 0 - 4095

1p 4096 - 8191

2p 8192 - 12287

3p 12288 - 16383

4p 16384 - 20479

Pe pagina 4 cu deplasamentul  
20000 – 16384 = 3616

Pentru pagina de 8 KB

Op 0 – 8191

1p 8192 – 16383

2p 16384 - 24575

Pe pagina 2 cu deplasamentul

20000 – 16384 = 3616

32768 - este p4 deplasamentul 0

60000 este p7 deplasamentul 2656

### Sarcini de realizare:

1. De exemplu, având 64 KB de spațiu de adrese virtuale și 32 KB de memorie fizică, cu mărimea paginii de 4KB obținem 16 pagini virtuale și 8 blocuri de pagini. Transferul de date între RAM și disc are loc întotdeauna în pagini. În memoria virtuală MV și în memoria operativă MO paginile se înscriu în ordinea indicată în Tab.1. De apreciat numărul paginii memoriei virtuale și a memoriei operative pentru adresele indicate în Tab.1

Tab.1

Var	Ordine paginilor în MV	Ordine paginilor în MO	Adrese de memorie
1	1,3,5,7,9,11,13,15, 0,2,4,6,8,10,12,14	0,5,2,10,14,7,15,9	41000,24500,10000,120
2	3,5,7,9,1,11,13,15, 4,6,8,2,10,12,14,0	1,5,9,3,0,11,4,7	32000,360,18000,57000
3	2,4,6,8,10,12,14,15, 0,1,3,5,7,9,11,13	14,0,8,5,12,10,1,2	25000,34000,11000,80
4	4,6,8,2,10,12,14,0, 1,3,5,7,9,11,13,15	3,1,5,9,12,11,10,8	21000,47000,5000,160
5	2,3,1,4,6,7,5,8, 10,11,9,13,15,14,0,12	10,9,8,6,5,0,2,1	30000,14000,43000,90
6	0,2,1,3,4,6,5,7,9, 8,10,12,11,15,13,14	8,4,10,1,12,5,3,15	18000,38000,9000,300
7	8,9,7,10,6,11,5,12, 4,13,3,14,2,15,0,1	6,3,9,15,2,10,8,5	21000,37000,8000,250
8	1,15,2,14,3,13,4,12, 5,11,6,10,7,0,8,9	7,12,1,9,6,3,10,8	44000,22000,15000,60
9	0,15,2,14,3,13,4,12, 5,11,6,10,7,9,8,1	14,5,0,9,13,7,2,12	5000,29000,24000,110
10	2,4,6,8,10,12,14,0,1,3,5, 7,9,11,13,15	3,11,0,14,5,2,8,12	19000,42000,13000,350
11	5,11,3,15,1,9,7,13,2,4,6,8,1 0,12,14,0	11,1,13,5,3,7,15,9	27000,31000,10000,180
12	13,9,5,1,15,11,7,3,0,2,4, 6,8,10,12,14	10,4,1,15,8,6,2,14	35000,17000,12000,450
13	6,14,2,10,4,12,0,8,1,9,3, 11,5,13,7,15	0,7,15,3,12,5,9,13	23000,41000,20000,130

14	9,3,7,11,15,1,5,13,0,2,4,6,8,10,12,14	2,9,6,1,14,3,11,5	32000,16000,45000,90
15	8,0,12,4,14,6,2,10,1,5,9,13,3,7,11,15	3,12,8,6,14,5,0,11	26000,33000,14000,190

Realizat pentru mărimi de pagini de 8KB din Tab.2

Tab.2.

Var	Ordine paginilor în MV	Ordine paginilor în MO	Adrese de memorie
1	0,2,4,6,1,3,5,7	4,7,0,2	30000,8015,62300,24500
2	1,3,5,7,0,2,4,6	1,6,5,4	80, 13900, 33888,65000
3	1,3,5,7,9,11,13,15	2,5,8,11	15000,27000,32000,45000
4	2,4,6,8,10,12,14,0	1,7,10,13	21000,34000,17000,10000
5	15,13,11,9,7,5,3,1	3,12,7,1	25000,30000,9000,18000
6	4,12,8,2,14,10,6,0	5,10,15,2	14000,43000,11000,5000
7	7,9,11,13,1,3,5,15	0,8,6,12	12000,22000,39000,5000
8	0,14,2,10,4,8,6,12	2,11,5,9	27000,14000,45000,18000
9	6,14,10,2,4,8,0,12	4,1,15,7	13000,37000,41000,30000
10	9,5,3,7,15,13,11,1	8,12,4,10	35000,25000,16000,42000
11	2,6,10,14,0,4,8,12	3,9,11,1	15000,27000,10000,38000
12	5,13,9,1,7,3,15,11	6,0,13,2	22000,34000,47000,9000
13	12,8,4,0,14,10,6,2	1,10,7,15	30000,19000,5000,32000
14	7,15,3,11,1,5,9,13	4,12,3,8	23000,11000,36000,45000
15	0,2,4,6,8,10,12,14	11,5,2,9	13000,25000,33000,7000

Realizat pentru mărimi de pagini de 2KB din Tab.3

Tab.3.

Var	Ordine paginilor în MV	Ordine paginilor în MO	Adrese de memorie
1	1,3,5,7,9,11,13,15,17,19,21,23,25,27,0,2,4,6,8,10,12,14,16,18,21,22,24,26	6,5,4,3,2,1,0,23,25,27,13,18	25555, 16666, 366, 19999
2	0,2,4,6,8,10,12,14,16,18,20,22,24,26,1,3,5,7,9,11,13,15,17,19,21,23,25,27	3,9,12,5,15,7,11,14,1,8,4,6	13000,27000,31000,49000
3	15,13,11,9,7,5,3,1,0,2,4,6,8,10,12,14,16,18,20,22,24,26,17,19,21,23,25,27	1,10,5,14,7,3,12,8,11,0,4,15	24000,32000,19000,15000
4	3,6,9,12,15,18,21,24,27,0,2,5,8,11,14,17,20,23,26,1,4,7,10,13,16,19,22,25	8,2,14,11,0,5,9,3,7,15,6,12	21000,38000,47000,9000
5	10,8,6,4,2,0,12,14,16,18,20,22,24,26,1,3,5,7,9,11,13,15,17,19,21,23,25,27	7,4,15,9,10,1,14,11,5,0,6,3	11000,33000,29000,17000
6	1,3,5,7,9,11,13,15,17,19,21,23,25,27,0,2,4,6,8,10,12,14,16,18,20,22,24,26	11,0,7,12,5,9,14,6,3,10,8,15	13000,27000,31000,45000
7	5,10,15,20,25,0,3,6,9,12,18,21,24,27,1,4,8,11,13,17,19,23,2,7,14,16,22,26	9,15,4,11,2,7,10,3,5,14,8,6	23000,34000,10000,36000

8	14,7,0,3,10,13,8,11,18,21,24,27,6,1,9,5,12,4,16,19,22,25,2,17,15,20,23,26	5,13,8,2,7,0,14,10,1,9,6,11	15000,29000,42000,12000
9	0,2,5,9,11,13,16,19,23,25,1,4,7,10,12,15,18,20,22,26,3,6,8,14,17,21,24,27	6,12,3,15,0,5,11,8,7,10,1,14	14000,37000,25000,43000
10	12,8,4,0,14,10,6,2,18,15,1,9,13,3,7,5,17,11,23,19,27,21,24,16,26,20,22,25	3,7,15,12,10,5,1,11,6,2,9,14	18000,31000,40000,15000
11	3,9,15,21,27,0,5,11,17,23,2,7,13,19,25,1,6,12,18,24,8,10,14,20,26,4,16,22	11,8,2,6,1,3,12,14,0,5,10,9	26000,17000,35000,8000
12	0,7,14,21,2,9,16,23,3,8,15,22,4,11,18,25,5,10,17,24,1,6,13,20,27,12,19,26	2,13,4,10,1,6,9,5,12,11,8,3	20000,14000,43000,32000
13	10,5,0,15,20,25,7,2,9,14,19,24,11,6,1,16,21,26,8,3,12,17,22,27,13,4,18,23	9,3,14,0,7,12,2,5,11,10,6,15	21000,30000,44000,13000
14	12,4,15,23,6,10,16,24,8,3,13,25,18,7,19,1,22,0,5,11,20,17,26,9,2,14,27,21	4,0,10,8,11,2,13,1,7,15,3,9	18000,33000,27000,5000
15	7,14,3,11,0,15,4,12,1,6,18,10,23,9,25,5,13,22,2,17,26,8,19,24,27,16,20,21	3,6,10,14,12,5,7,11,0,15,8,2	28000,40000,14000,24000

2. Sunt date la realizare 12 procese cu timpul de realizare din Tab.4. Lista intervalelor de memorie liberă indicate în Tab.4. Apreciați cel mai optimal și cel mai neoptimal algoritmul din următoarele algoritme:

- Algoritmul primul interval potrivit.
- Algoritmul cel mai potrivit interval
- Algoritmul următorul interval potrivit
- Algoritmul cea mai nepotrivit interval

Motivați răspunsul.

Tab.4.

<b>Va r</b>	<b>Volumul memoriei operative pentru fiecare proces</b>	<b>Lista intervalelor libere a MO</b>
1	21,9,28,15,45,40,10,25,15,30,15,28	20,25,45,50,10,16,50,18,20,30,22,12,42,18,30,29
2	30,12,27,17,35,41,13,24,14,28,18,25	18,22,46,51,11,14,52,19,25,31,24,15,40,21,29,30
3	25,14,29,16,42,33,11,28,18,32,20,30	23,26,40,53,12,19,48,15,22,27,21,10,39,16,31,28
4	22,10,26,18,34,36,15,29,12,31,21,27	24,29,41,48,13,15,50,18,20,28,23,14,38,17,33,26
5	24,11,31,19,37,42,17,22,13,35,16,28	21,27,44,49,10,18,47,20,23,26,19,12,45,22,32,25
6	28,13,34,21,39,31,14,26,19,30,23,32	22,30,43,55,12,16,49,14,21,29,25,13,41,19,34,27
7	27,15,33,20,38,35,12,30,17,29,24,28	25,28,45,50,10,19,53,15,26,33,20,13,44,18,36,24

8	29, 17, 32, 18, 40, 37, 16, 27, 20, 34, 22, 25	23, 26, 42, 52, 11, 14, 51, 17, 25, 30, 22, 15, 39, 20, 33, 29
9	31, 16, 30, 22, 41, 33, 13, 24, 21, 36, 19, 27	19, 27, 46, 54, 10, 17, 49, 16, 24, 28, 23, 12, 42, 18, 32, 26
10	26, 12, 28, 19, 37, 39, 11, 29, 15, 35, 18, 31	21, 25, 47, 53, 13, 16, 50, 14, 22, 32, 19, 11, 40, 20, 34, 23
11	33, 18, 29, 15, 42, 38, 14, 26, 17, 31, 22, 24	20, 26, 41, 56, 12, 15, 48, 13, 23, 29, 25, 14, 43, 21, 35, 27
12	30, 14, 27, 16, 39, 34, 12, 25, 21, 33, 19, 28	18, 24, 49, 54, 10, 17, 52, 16, 20, 31, 26, 13, 46, 22, 38, 23
13	28, 16, 35, 19, 41, 32, 15, 29, 18, 36, 20, 30	22, 27, 44, 51, 11, 13, 45, 14, 21, 33, 24, 15, 47, 19, 31, 25
14	25, 12, 31, 21, 37, 39, 17, 28, 13, 32, 23, 27	26, 29, 42, 55, 10, 18, 48, 20, 24, 30, 22, 14, 41, 15, 34, 28
15	32, 13, 30, 18, 38, 35, 11, 26, 19, 33, 21, 29	24, 28, 47, 50, 12, 16, 53, 14, 23, 27, 20, 11, 44, 17, 36, 26

3. În memoria operativă sunt încărcate 12 pagini după ordine de la 0,1,2,... la 12. Volumul memoriei operative pentru fiecare proces este indicat în Tab.5. Lista intervalelor libere a memoriei operative din Tab.5. De apreciat la cererea de întrerupere a paginii care pagină trebuie exclusă din memoria operativă pentru a înscrie o nouă pagină pentru următorii algoritmi?

- Algoritmul FIFO
- Algoritmul a doua încercare
- Algoritmul NRU
- Algoritmul LRU

Tab.5.

Var	Numărul paginii	Timpul de încărcare	Timpul ultimului apel	Bitul R	Bitul M
1	0 - 12	1,3,5,7,9,4,6,11,10,2,4,8, 12	15,17,20,22,24,19,21,26,25,31,33,38,16	1,1,0,0,0,1,1,0,1,0,1,1,1	0,0,1,1,0,1,1,0,0,0,1,0
2	0 - 12	2, 4, 6, 8, 10, 3, 7, 9, 11, 1, 5, 12, 4	13, 18, 21, 23, 25, 17, 19, 29, 30, 34, 35, 39, 15	0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0	1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0
3	0 - 12	1, 3, 7, 6, 2, 8, 10, 4, 11, 5, 9, 12, 4	12, 16, 23, 25, 20, 30, 32, 28, 34, 36, 29, 37, 18	1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1	0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0
4	0 - 12	4, 7, 3, 5, 1, 10, 6, 9, 11, 8, 2, 12, 4	14, 17, 27, 30, 22, 19, 31, 29, 28, 33, 38, 35, 16	0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1	1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1
5	0 - 12	6, 2, 4, 8, 10, 5, 1, 7, 12, 3, 9, 11, 4	16, 20, 26, 31, 23, 33, 27, 32, 36, 28, 29, 38, 15	1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1	0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0
6	0 - 12	2, 8, 5, 3, 10, 7, 1, 12, 4, 6, 9, 11, 4	13, 18, 27, 24, 28, 32, 29, 35, 36, 31, 39, 21, 15	0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1	1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0
7	0 - 12	5, 2, 8, 6, 1, 10, 3, 12, 4, 7, 11, 9, 4	15, 21, 24, 19, 25, 34, 28, 30, 33, 35, 29, 36, 14	1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1	1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1
8	0 - 12	6, 4, 2, 8, 7, 10, 5, 9, 12, 1, 11, 3, 4	16, 20, 25, 28, 30, 33, 26, 31, 37, 22, 39, 35, 15	1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0	1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0

9	0 - 12	7, 1, 5, 3, 9, 8, 2, 10, 12, 4, 6, 11, 4	13, 18, 27, 22, 25, 30, 29, 34, 36, 32, 38, 28, 17	0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1	1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0
10	0 - 12	8, 2, 6, 3, 7, 10, 5, 12, 4, 1, 9, 11, 4	14, 19, 26, 24, 31, 28, 32, 30, 34, 23, 29, 37, 16	1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1	0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0
11	0 - 12	3, 7, 2, 8, 5, 6, 9, 12, 1, 4, 11, 10, 4	15, 20, 27, 31, 24, 28, 33, 36, 32, 34, 39, 30, 17	1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1	0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0
12	0 - 12	9, 5, 2, 6, 4, 8, 1, 12, 10, 7, 3, 11, 4	16, 21, 24, 28, 26, 30, 32, 35, 33, 37, 31, 29, 14	1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0	1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1
13	0 - 12	4, 6, 3, 9, 2, 8, 5, 1, 11, 7, 12, 10, 4	13, 19, 25, 27, 31, 29, 33, 24, 37, 35, 30, 28, 16	0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1	1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0
14	0 - 12	7, 1, 5, 2, 10, 8, 3, 12, 4, 9, 6, 11, 4	12, 18, 21, 27, 24, 28, 35, 33, 29, 30, 38, 25, 15	1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0	0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1