

Лабораторная работа №6

Планирование оперативной памяти

Если процессы и свободные участки памяти хранятся в списке, существует несколько алгоритмов предоставления памяти процессу:

1. **Алгоритм «первый подходящий участок».** Работает следующим образом: менеджер памяти просматривает список участков, пока не находит достаточно большой свободный участок. Этот участок затем делится на две части: одна дается процессу, а другая остается неиспользованной. Это происходит всегда, за исключением статистически нереалистичного случая точного соответствия размера свободного участка и процесса. Это быстрый алгоритм, поскольку поиск максимально сокращается.
2. **Алгоритм «следующий подходящий участок»** работает так же, как и первый алгоритм, но каждый раз, когда он находит подходящий свободный участок, он запоминает адрес, где остановился. И в следующий раз, когда вызывается алгоритм, он начинает с того же места, где остановился в прошлый раз, вместо того чтобы начинать поиск с начала списка.
3. **Алгоритм «наиболее подходящий участок»** ищет и выбирает наиболее близкий по размеру участок. Вместо того чтобы делить большой нераспределенный участок, который может понадобиться позже, этот алгоритм пытается найти участок, максимально приближенный к реально необходимому размеру.
4. **Алгоритм «наименее подходящий участок»** всегда выбирает самый большой свободный участок, из которого после разделения останется участок достаточного размера, который можно будет использовать в будущем.

Задано: 5 процессов с объемом необходимой оперативной памяти:

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

Список свободных участков памяти:

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

Распределение памяти

Для алгоритма «первый подходящий»:

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)

Итоговый список: L(10, 45, 26, 6, 18, 20, 32)

157

Для алгоритма «следующий подходящий»:

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)

Итоговый список: L(17, 10, 60, 16, 6, 18, 32)

159

Для алгоритма «наиболее подходящий»:

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)

Итоговый список: L(40, 65, 6, 10, 32)

153

Для алгоритма «наименее подходящий»:

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)

Итоговый список: L(12, 25, 20, 26, 6, 18, 20, 32)

159

Оптимальным будет **алгоритм «наименее подходящий»**. Он имеет то же время использования, что и «следующий подходящий», но оставляет больше пригодных свободных участков.

Алгоритмы замены страниц

Когда возникает ошибка страницы, операционная система должна выбрать страницу для удаления из памяти, чтобы освободить место для страницы, которую нужно будет загрузить в память. Если удаляемая страница была изменена, пока находилась в памяти, её необходимо записать на диск, чтобы обновить хранящуюся там копию. Однако если страница не была изменена (например, содержит текст программы), то копия на диске уже является самой актуальной и не нуждается в перезаписи. Тогда страница просто считывается поверх удаляемой страницы.

1. Алгоритм FIFO (First-In, First-Out — первым пришёл, первым ушёл).

Операционная система поддерживает список всех страниц, находящихся в данный момент в памяти, где первая страница — самая старая, а страницы в конце списка — самые новые. Когда возникает ошибка страницы, страница из начала списка удаляется из памяти, и новая страница добавляется в конец списка.

Если использовать алгоритм FIFO в магазине, он может удалить воск для усов, но также может удалить муку, соль или масло.

2. Алгоритм второй попытки — это алгоритм FIFO, который избегает проблемы удаления часто используемых страниц из памяти. Проверяется бит R самой старой страницы. Если он равен 0, страница не только долго находится в памяти, но и не используется, поэтому она немедленно заменяется новой. Если бит R равен 1, ему присваивается значение 0, страница перемещается в конец списка, и её время загрузки обновляется, то есть страница считается только что загруженной в память. Процедура затем продолжается.

3. Алгоритм NRU (Not Recently Used — не использовалась недавно) удаляет самую старую страницу, используя случайный выбор из непустого класса с наименьшим номером.

Когда возникает ошибка страницы, операционная система проверяет все страницы и делит их на четыре категории на основе текущих значений битов R и M:

- Класс 0: без обращений и без изменений.
- Класс 1: без обращений, страница была изменена.
- Класс 2: были обращения, страница не была изменена.
- Класс 3: были как обращения, так и изменения.

4. Алгоритм LRU (Least Recently Used — наименее недавно использованная) — удаляется страница, которая дольше всех не использовалась.

Когда происходит ошибка страницы, удаляется страница, которая не

использовалась дольше всего. Эта стратегия замены страниц похожа на алгоритм NRU, но удаляется не самая старая страница, а именно та, которая была дольше всего неактивна.

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:

Алгоритм FIFO

Будет заменена страница 3, так как она самая старая и имеет время загрузки 110.

Алгоритм второй попытки

Будет заменена страница 0, так как она самая старая с битом R=0. Страница 3 — старше, но у неё бит R=1.

Алгоритм NRU

Будет заменена страница 0, так как она самая старая с битами R=0 и M=0.

Алгоритм LRU

Будет заменена страница 2, так как у неё наименьшее значение времени последнего обращения — 270, а также R=0 и M=0.

Организация памяти страниц

Виртуальное адресное пространство делится на единицы, называемые **страницами**. Соответствующие единицы в физической памяти называются **блоками страниц**. Страницы и соответствующие им блоки всегда имеют одинаковый размер. В реальных системах используются размеры страниц по **512 байт**.

На какой виртуальной странице будет находиться физический адрес 20000 и каково будет смещение?

Для страниц размером 4 КБ:

Страница 0: 0 – 4095

Страница 1: 4096 – 8191

Страница 2: 8192 – 12287

Страница 3: 12288 – 16383

Страница 4: 16384 – 20479

→ На странице 4 с смещением:

$$20000 - 16384 = 3616$$

Для страниц размером 8 КБ:

Страница 0: 0 – 8191

Страница 1: 8192 – 16383

Страница 2: 16384 – 24575

→ На странице 2 с смещением:

$$20000 - 16384 = 3616$$

Примеры:

- **32768** — это страница 4 со смещением **0**
- **60000** — это страница 7 со смещением **2656**

Задания к выполнению:

1. Например, имея **64 КБ** виртуального адресного пространства и **32 КБ** физической памяти, при размере страницы **4 КБ** получаем **16 виртуальных страниц** и **8 блоков страниц**.

Передача данных между RAM и диском всегда осуществляется постранично. В виртуальной памяти (MV) и в оперативной памяти (MO) страницы записываются в порядке, указанном в Таблице 1.

Необходимо определить **номер страницы в виртуальной памяти** и **в оперативной памяти** для адресов, указанных в Таблице 1.

Таблица 1

Ва р.	Порядок страниц в MV (вирт. память)	Порядок страниц в МО (опер. память)	Адреса памяти
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, 10, 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

Реализация для размера страниц 8 КБ из Таблицы 2

Таблица 2

Вариант	Порядок страниц в виртуальной памяти (MV)	Порядок страниц в оперативной памяти (MO)	Адреса памяти
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

Реализация для размера страниц 2 КБ из Таблицы 3

Таблица 3

Ва р.	Порядок страниц в виртуальной памяти (MV)	Порядок страниц в оперативной памяти (MO)	Адреса памяти
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	3,9,15,21,27,0,5,11,17,23,2,7,13,19,2 5,1,6,12,18,24,8,10,14,20,26,4,16,22	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	3,6,10,14,12,5,7,11,0,1 5,8,2	28000, 40000, 14000, 24000

2. Даны 12 процессов с объемом оперативной памяти для каждого процесса. Список свободных интервалов памяти в оперативной памяти. Оцените самый оптимальный и наименее оптимальный алгоритм из следующих:

- Алгоритм первого подходящего интервала.
- Алгоритм наиболее подходящего интервала.
- Алгоритм следующего подходящего интервала.
- Алгоритм наименее подходящего интервала.

Таблица 4:

Вар.	Объем оперативной памяти для каждого процесса	Список свободных интервалов памяти в МО
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. В оперативной памяти загружены 12 страниц по порядку от 0,1,2,... до 12. Объем оперативной памяти для каждого процесса указан в Таблице 5. Список свободных интервалов памяти в оперативной памяти приведен в Таблице 5. Оцените, какая страница должна быть исключена из оперативной памяти при запросе на прерывание страницы для загрузки новой страницы для следующих алгоритмов:

- Алгоритм FIFO
- Алгоритм второй попытки
- Алгоритм NRU
- Алгоритм LRU

Таблица 5

Вариант	Номер страницы	Время загрузки	Время последнего обращения	Бит R	Бит M
1	0 - 12	1,3,5,7,9,4,6,1 1,10,2,4,8,12	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, 1,0,0,0,0,1, 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, 0, 0, 1, 1, 1, 0, 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, 1, 1, 1, 0, 0, 1