# ASSESSMENT II - Model answer

Consider a disk queue with I/O requests on the following cylinders in their arriving order:

6, 10, 12, 54, 97, 73, 128, 15, 44, 110, 34, 45

The disk head is assumed to be at Cylinder 23 and moving in the direction of decreasing number of cylinders.

The disk consists of a total of 150 cylinders. Calculate the **total disk head movement** using the **LOOK**-scheduling algorithm.

Solution
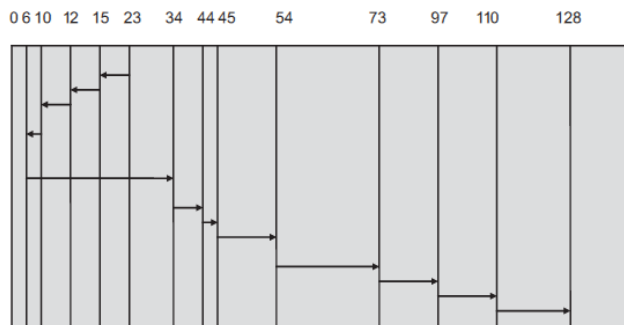
The total head movement = |23-6| + |128-6| = 17 + 122 = 139

Consider four memory partitions of size **400 KB, 600 KB, 500 KB** and **250 KB**. These partitions need to be allocated to four processes of sizes:

Process P1 = 357 KB

Process P2 = 210 KB

Process P3 = 468 KB

Process P4 = 491 KB

in that order.

Perform the allocation of processes using the **Best Fit Algorithm**.

 If, in a dynamic partition memory management system, the current value of the base register is **42993** and the current value of the limit register is **2031**, compute the physical addresses that correspond to the following logical addresses:
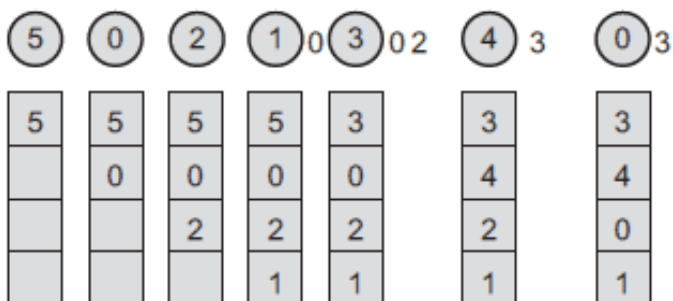
1755

A system uses **4 page frames** for storing process pages in main memory.

It uses the **First In First Out (FIFO)** page replacement policy. Assume that all the page frames are initially empty.

What is the **total number of page faults** that will occur while processing the page reference string given below?

5 0 2 1 0 3 0 2 4 3 0 3

To detect errors in a hard disk block, we need to count the number of **1**'s. The block contains binary data. Binary data is a type of data that only has two possible values (0 and 1).

The frame contains only binary data. Binary data is a type of data that only has two possible values (0 and 1).

Complete the skeleton code of a program that is trying to count how many digits of a series of binary data that are a **1**.

To help you with this task, please write a function called **countOnes** that computes the number of digits of an input binary data that are equal to **1**. Be sure to also write a prototype for this function.

Next, please use **countOnes** in your main function. The first input is how many series of binary data, followed by the strings (see the test cases).

Display **the total count of ones**.

Test case 1:

Input:
2
00011100 110

Output:
5

Test case 2:

Input:
3
00011100 110 111

Output:
8

Test case 3:

Input:
1
000000

Output:
0

Solution

```c
#include <stdio.h>
int countOnes(int);
int main() {
   int i, num, number, total = 0;
   scanf("%d", &num);
   for (i=0; i<num; i++) {
      scanf("%d", &number);
      total += countOnes(number);
   }
   printf("%d\n", total);
   return 0;
}
int countOnes(int number) {
   int count = 0;
   while (number > 0) {
      if (number % 10 == 1)
         count++;
      number = number/10;
   }
   return count;
}
```