

## HEXAWARE – DAY 1

A bear starting from the point P, walked one mile due south,  
Then he changed direction and walked one mile due east. Then he  
turned again to the left and walked one mile due north and arrived at point P he started from  
what was the color of the bear?

Ans : The bear is **white** because this scenario can only occur at the **North Pole** where  
walking one mile south, one mile east, and one mile north returns to the starting point.

2. Two towns A and B are 3 km s apart It is proposed to build  
a new school serving 100 students in town A and 50 students  
in town B. How far from town A should the school be built  
if the total travel distance by all 150 students is to be as small as  
possible?

1. **Start**
2. **Input** the number of students in Town A ( $n_1 = 100$ ) and Town B ( $n_2 = 50$ )
3. **Input** the total distance between the towns ( $d = 3$  km)
4. Divide the distance in the **reverse ratio** of the number of students:  
$$N_2/n_2+n_1// d$$
5. **Display** the calculated distance x
6. **End**

3. A traveller arrives at hotel he has no money but only a silver chain consisting of 6 links.  
He uses one link to pay for each day

spent at the hotel but the hotel manager agrees to accept no more than one broken link

How should the traveller cut up the chain in order to settle the  
amount with the hotel manager on a daily basis

1. what is the least number of links that have to be cut if the traveller stays 100 days at  
the hotel and has a chain consisting of 100 links? what is the answer in general case n  
days and n links

### 1. Start

2. **Input** the total number of links in the chain ( $n = 100$ )
3. **Initialize** cuts = 0
4. While  $n > 0$ :
5. Cut links in powers of 2: 1, 2, 4, 8, 16, ... (smallest combinations to sum up to  $n$ )
6. After each cut, assign those segments accordingly to match required payments
7. Increment cuts by 1 for each cut made
8. Continue until the chain is divided in segments that can make all payments
9. **Display** the minimum number of cuts required
10. **End**

4. Rearrange the letters in the words new door to make one word:

1. **Start**
2. **Input** the given string "new door"
3. **Remove** spaces from the string
4. **Generate all possible permutations** of the characters
5. For each permutation:  
→ **Check** if the permutation is a valid English word
6. If a valid word is found:
7. **Display** the word
8. If no valid word is found:
9. **Display** "No valid word found"
10. **End**

5. do divide and conquer 6 5 1 4 3 2

1. **Start**
2. If the list has **one or zero** elements, return the list (base case)
3. Divide the list into two halves:
4. **Left Half** = First half of the list
5. **Right Half** = Second half of the list
6. Recursively apply **Merge Sort** on both halves
7. Merge the two sorted halves by:
8. Initialize two pointers for each half
9. Compare elements from both halves
10. Add the smaller element to the merged list
11. Continue until all elements are merged
12. Return the merged sorted list
13. **End**

6, Draw flowchart for calculating simple interest

1. **Start**
2. **Input** Principal (P), Rate of Interest (R), and Time Period (T)
3. **Initialize** Simple Interest (SI) = 0
4. **Calculate**  $SI = (P \times R \times T) / 100$
5. **Display** SI

6. End

By,  
Divyalakshmi M