**1D Array and Passing 1D array to a function**

**Basics of 1D array (Declaration and Initialization)**

//Arrays

#include<stdio.h>

int main()

{

int n,i;

int arr[]={1,2,3};//Size of this array is 3

int arr1[2]={1,2};

int arr2[5]={0};//All elements will be initialized to zero

int arr3[5]={1};//Only first element will be initialized to 1 and rest will be zeros

int arr4[100];

int arr5[5]={};//All elements will be initialized to zero

printf("%d",arr[0]);

printf("\n%d",arr1[1]);

printf("\n%d",arr5[3]);

printf("\n%d",sizeof(arr1));

printf("\n%u %u %u %u",&arr[0],&arr[1],arr,&arr);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Taking input from user

printf("\n Enter size of array(Less than or equal to 100):");

scanf("%d",&n);

printf("\n Enter array elements:");

for(i=0;i<n;i++)

{

scanf("%d",&arr4[i]);

}

printf("\n Array elements are:");

for(i=0;i<n;i++)

{

printf("\n%d",arr4[i]);

}

return 0;

}

**Write a program to read and display elements of 1D array**

#include<stdio.h>

int main()

{

int a[100],n,i;

printf("\n Enter number of elements:");

scanf("%d",&n);

printf("\n Enter array elements:");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

printf("\n Entered array elements are:");

for(i=0;i<n;i++)

{

printf("\n%d",a[i]);

}

return 0;

}

**Write a program to find the sum of all elements of 1D array**

#include<stdio.h>

int main()

{

int a[100],n,i,sum=0;

printf("\n Enter number of elements:");

scanf("%d",&n);

printf("\n Enter array elements:");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

for(i=0;i<n;i++)

{

sum=sum+a[i];

}

printf("\n Sum of array elements is:%d",sum);

return 0;

}

**Write a program to display the largest and smallest element from the group of 1D array elements**

#include<stdio.h>

int main()

{

int n,a[10],i,max,min;

printf("\n Enter number of elements:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

min=a[0];

for(i=1;i<n;i++)

{

if(a[i]<min)

{

min=a[i];

}

}

max=a[0];

for(i=1;i<n;i++)

{

if(a[i]>max)

{

max=a[i];

}

}

printf("\nMaximum element is: %d",max);

printf("\nMinimum element is: %d",min);

return 0;

}

**Write a program to implement linear search in 1D array**

#include <stdio.h>

int main()

{

int a[50];

int i, loc = -1, key,n;

printf("\n Enter value of n:");

scanf("%d",&n);

printf("\n Enter the elements:");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

printf("Enter integer value to search in array:");

scanf( "%d", &key );

// attempt to locate searchKey in array a

for ( i = 0; i < n; i++ )

{

if ( a[i] == key )

{

loc = i; // location of key is stored

break;

} // end if

} // end for

if(loc!= -1)

{

printf("Element found at %d",loc+1);

}

else

{

printf("Element not found");

}

} // end main

**Write a program to implement binary search in 1D array**

#include<stdio.h>

int main()

{

int a[50],n,loc=-1, key, beg,last,mid,i;

printf("\n Enter number of array elements:");

scanf("%d",&n);

printf("\n Enter array elements:");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

beg=0;

last=n-1;

printf("Enter integer value to search in sorted array:");

scanf( "%d", &key );

while(beg<=last)//Loop will run until unless only one element is not remaining

{

mid = (beg + last) / 2; // determine index of middle element

if(a[mid]==key)

{

loc=mid; //save the location of element.

break;

}

else if(a[mid]>key) //Middle element is greater than key

{

last=mid-1;//If middle element is greater than key, we need to search left subarray

}

else if(a[mid]<key) //Middle element is less than key

{

beg=mid+1;//If middle element is less than key, we need to search right subarray

} //end of if else

} //end of while

if(loc!=-1)

{

printf("element found at %d", loc+1);//Location is exact position, not index

}

else

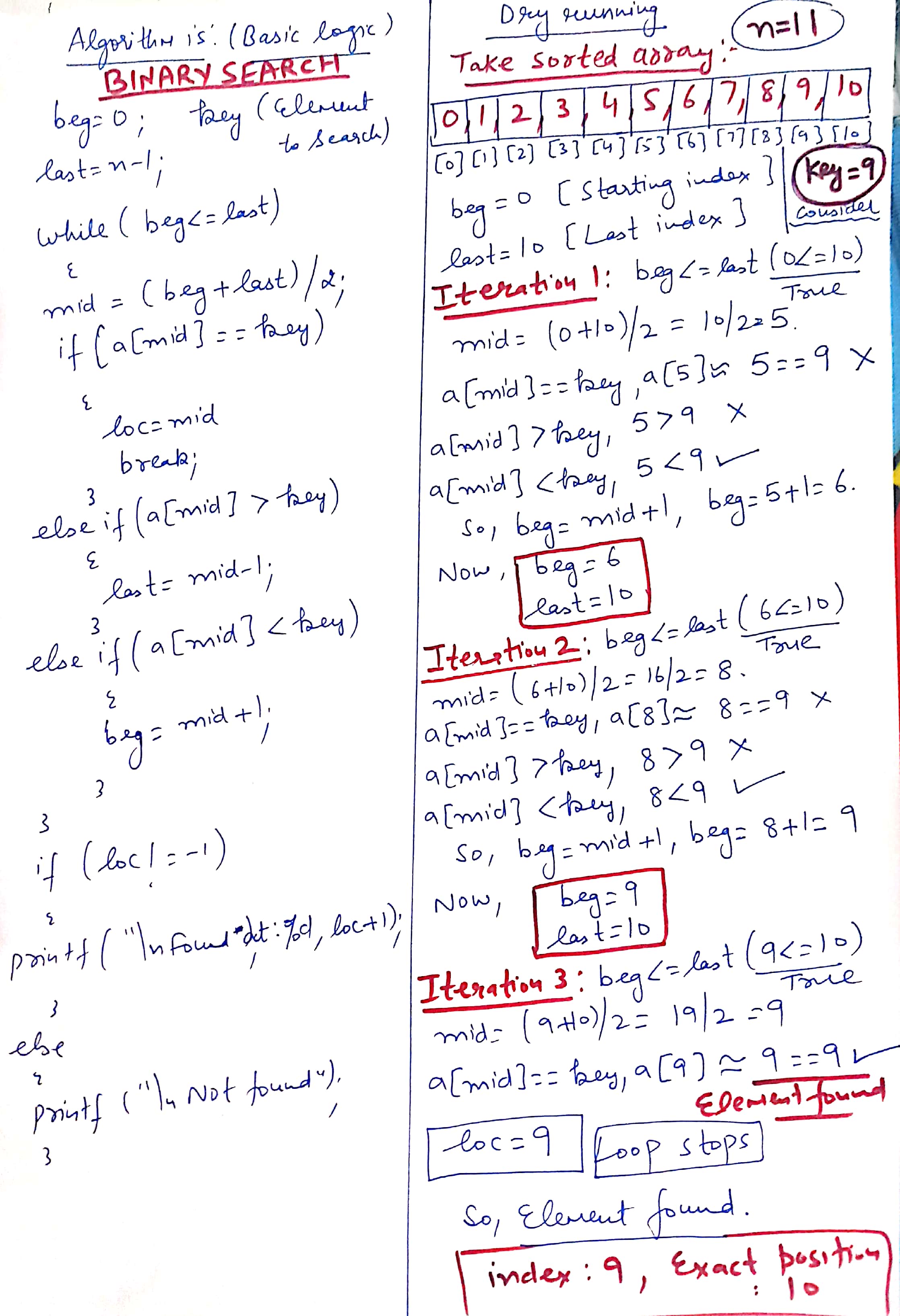
{

printf("element not found");

}

return 0;

}

**Dry running (Binary Search)**

**Write a program to sort the elements of an array in ascending order using bubble sort**

#include <stdio.h>

int main()

{

int a[100];

int hold,i,j,n;

printf("\n Enter value of n:");

scanf("%d",&n);

printf("\n Enter elements:");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

printf( "Data items in original order" );

for (i=0;i<n;i++ )

{

printf("%d ",a[i]);//Elements will come with space

} // end for

// bubble sort

// loop to control number of passes(no. of passes are always n-1)

for (i=0;i<n-1;i++)

{

// loop to control number of comparisons per pass(There is one comparison less)

for (j=0;j<n-i-1;j++)

{

// compare adjacent elements and swap them if first

// element is greater than second element

if (a[j]>a[j+1])

{

hold=a[j];

a[j]=a[j+1];

a[j+1]=hold;

} // end if

} // end inner for

} // end outer for

printf( "\nData items in ascending order" );

for (i=0;i<n;i++)

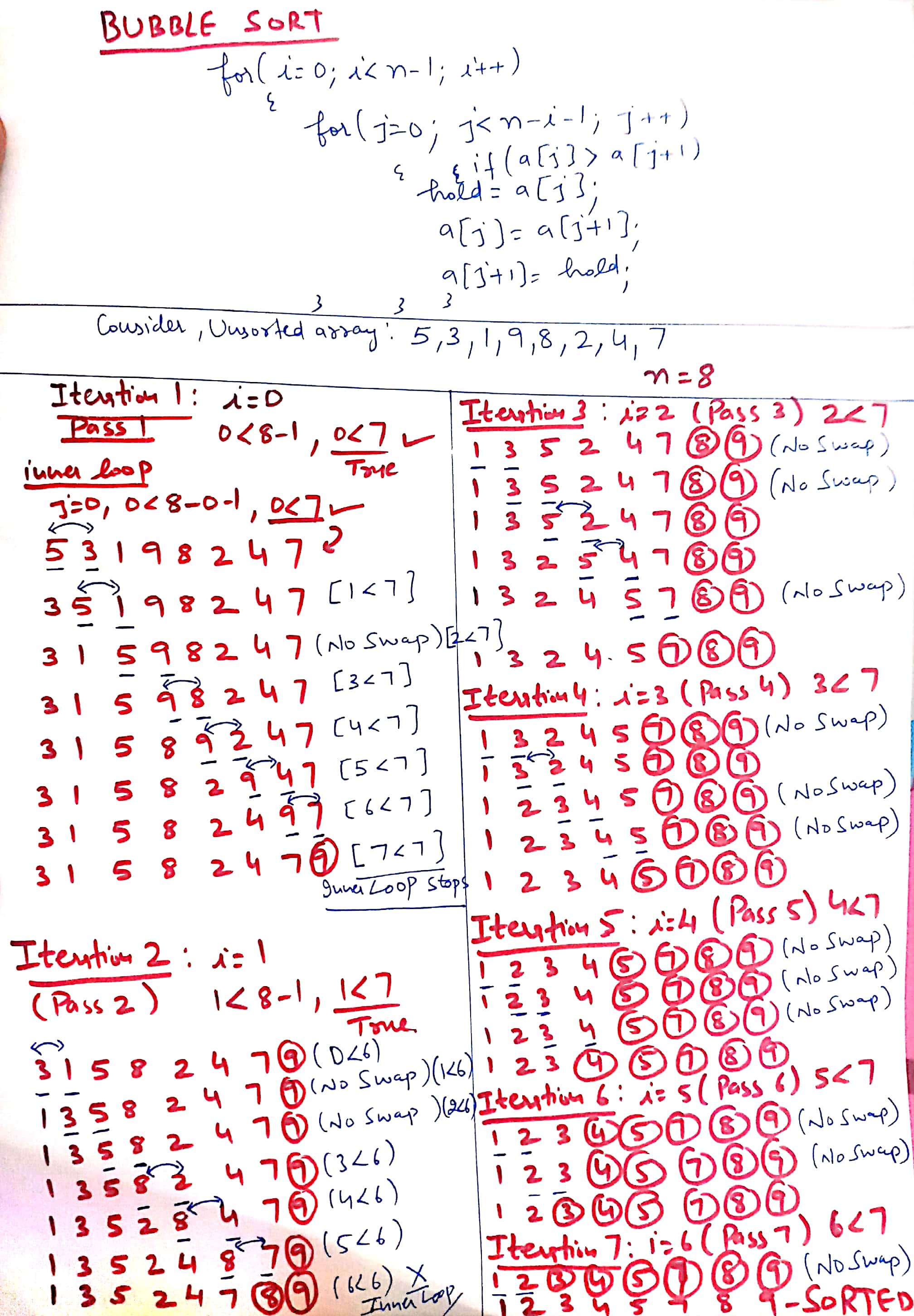
{

printf("%d ",a[i]);

} // end for

} // end main

**Dry running (Bubble sort)**

****

**Write a program insert an element at a given position in 1D array**

#include <stdio.h>

int main()

{

int array[100], position, c, n, value;

printf("Enter number of elements in array:\n");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

for (c = 0; c < n; c++)

{

scanf("%d", &array[c]);

}

printf("Enter the location where you wish to insert an element:\n");

scanf("%d", &position);

printf("Enter the value to insert:\n");

scanf("%d", &value);

for (c = n - 1; c >= position - 1; c--)

{

array[c+1] = array[c];

}

array[position-1] = value;

printf("Resultant array is:\n");

for (c = 0; c <= n; c++)

{

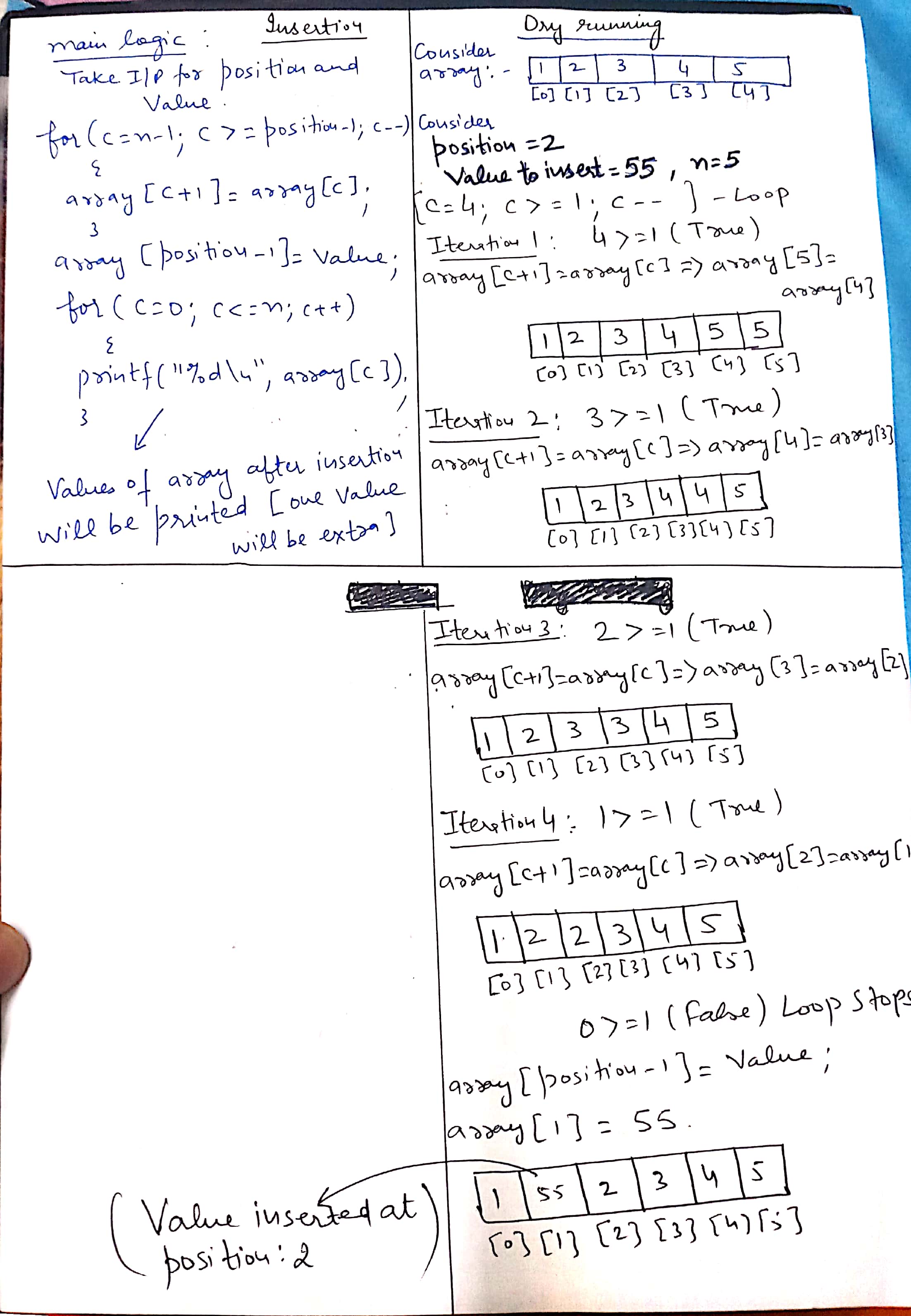
printf("%d\n", array[c]);

}

return 0;

}

**Dry running (Inserting element in array)**

****

**Write a program delete an element from a given position in 1D array**

#include <stdio.h>

int main()

{

int array[100], position, c, n;

printf("Enter number of elements in array\n");

scanf("%d", &n);

printf("Enter %d elements\n", n);

for (c = 0; c < n; c++)

{

scanf("%d", &array[c]);

}

printf("Enter the location where you wish to delete from an array\n");

scanf("%d", &position);

for (c = position-1; c < n-1; c++)

{

array[c] = array[c+1];

}

printf("Resultant array is\n");

for (c = 0; c < n-1; c++)

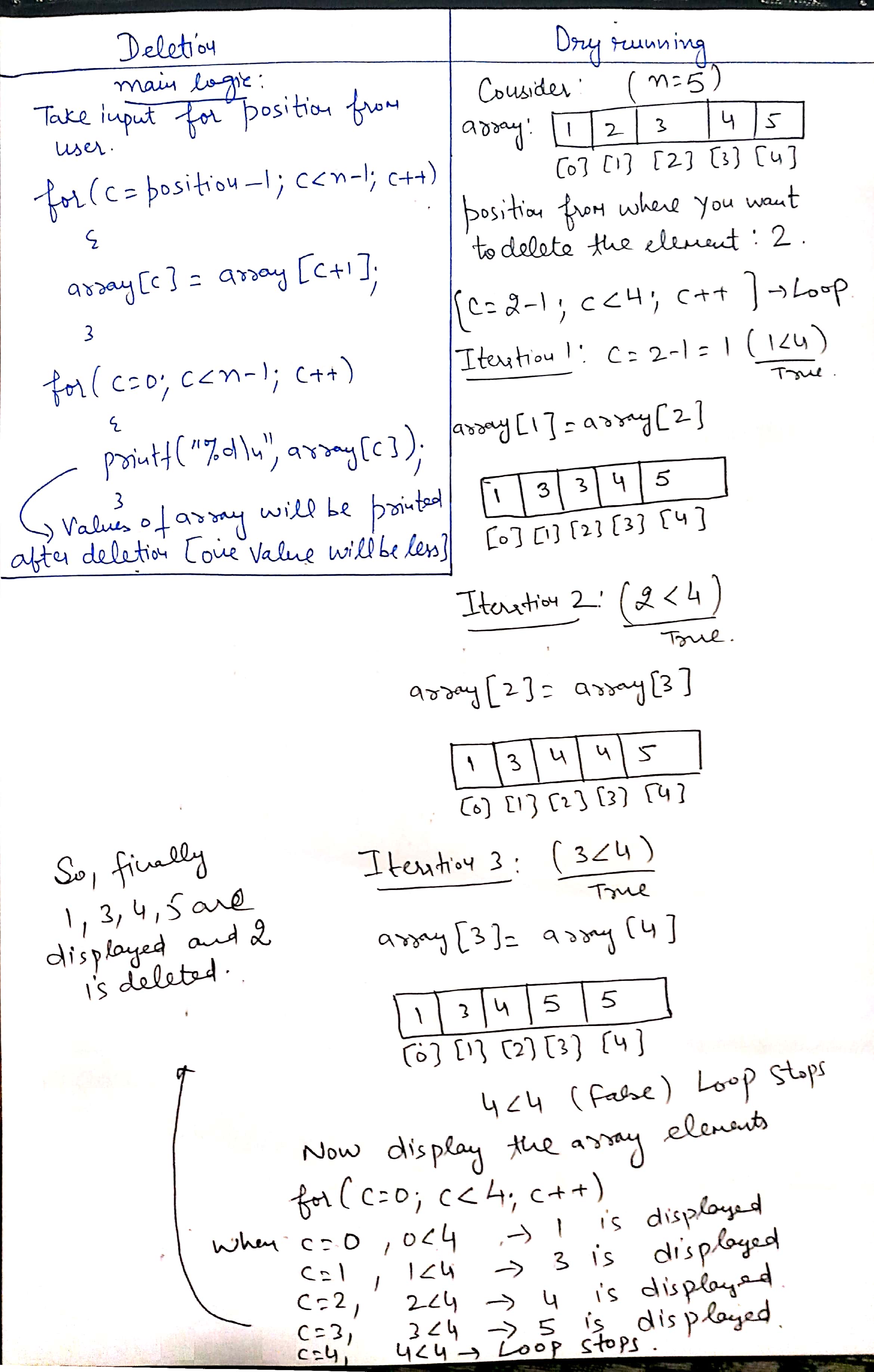
{

printf("%d\n", array[c]);

}

}

**Dry running (Deleting element)**

****

**Write a program to pass array to a function using By reference and By value approaches**

#include<stdio.h>

void reference(int[],int);

void value(int);

int main()

{

int arr[]={1,2,3,4,5};

int i;

printf("\n Elements by reference:");

reference(arr,5);//Passing array by call by reference

printf("\n");

printf("\n Elements by value:");

for(i=0;i<5;i++)

{

value(arr[i]);//Passing array value by Call by value

}

return 0;

}

void reference(int x[],int size)

{

int i;

for(i=0;i<size;i++)

{

printf("%d ",x[i]);

}

}

void value(int u)

{

printf("%d ",u);

}

**Passing array to a function-Example 2**

#include<stdio.h>

void reference(int[],int);

void value(int);

int main()

{

int arr1[]={1,2,3,4,5};

int arr2[]={12,13,14,15,16};

int i;

reference(arr1,5);//Passing array by call by reference

printf("\nElements of array after passing by reference to function:");

for(i=0;i<5;i++)

{

printf("\n%d",arr1[i]);//All elements are multiplied by 5(Changes are reflected)

}

printf("\n");

for(i=0;i<5;i++)

{

value(arr2[i]);//Passing array value by Call by value

}

printf("\nElements by array after passing by value to function:");

for(i=0;i<5;i++)

{

printf("\n%d",arr2[i]);// No changes are reflected as work was done on duplicate copy

}

return 0;

}

void reference(int x[],int size)

{

int i;

for(i=0;i<size;i++)

{

x[i]=x[i]\*5;

}

}

void value(int u)

{

u=u\*5;

}

**Practice questions to do:**

WAP to display the count of odd array elements

WAP to sort the array elements in descending order

WAP to find the sum of all prime array elements

WAP to perform linear search in an array after passing array to a function

WAP to display the second largest element from array

WAP to find the sum of those array elements which are multiple of 5

WAP to find the sum of two 1D array elements

WAP to display the elements of 1D array in reverse order