Name: Ahmal Afridi

I:D : 14226

SEMS : Summer.

SUBJECT:. Data Structure & algorithm.

ID # 14226 AKMAL AFRIDI (DSA)

Q1

Solution

1. First get the element to be Inserted Say x.
2. Then get the Position at which this element is to be Inserted.
3. Then Shift the array elements from this Position to one Position Forward & do this for all the other elements next to Position.
4. Insert the element x now at the Position Pos, as this is now empty.

Programming

```
Include <studio.h>
Int main ()
{
  Int arr[100] = {0}
  Int i, x, Pos, n = 10;
  // Initial array of Size 10
for (i= 0; i < 10; i++)
 arr[i] = i+1;
```

// Print the original array
for (i = 0; i < n; i++)

Printf ("%d", arr[i]);
Print f ("\n");
End

Q18

Solution Required

m:A (A) [52] = ?

Given

Base (A) = 54360 (lb & w=

w = 2

k = 52

lb = 1

P. T-6

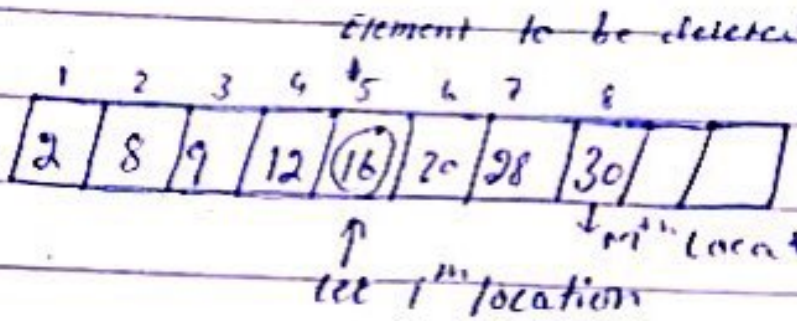
$$A(A[52]) = 54360 + 2(52-1)$$

$$= 54360 + 2(51)$$

$$= 54360 + 102$$

$$\boxed{= 54462}$$

Q2



Step 1. Back = i ⇒ Back 5

2. While (Back < r-1) repeat step 3 to 4

⇒ 5 < 8 true

3. Reg [Back] = Reg [Back + 1]

Reg [5] ⇒ Reg [6] ⇒ Reg [5] = 20

4. Back = Back + 1

Now Back = 6

Again Step 3

Reg [6] = Reg [6+1]

= 28

Step 4

Back = Back + 1 ⇒ 7

Again Step 3 until Back < r-1

Step 5 r1 = r1 - 1

= 8 - 1 ⇒ 7.

Step 6 End.

Q3

* C Program to Implement binary Search

```
Int binary search (int arr[],
Int l4 , Int r , Int x)
{
 17 (r >= 1) {
    Int mid = 1 + (r-1) /2 ;
17 (arr [mid] == x)
    return mid;
17 (arr [mid] > x)
return binary search (arr, 1, mid-1, x) ;
}return binary Search (arr, & mid + 1 , x, x);
return -1;
}

Int main (void)
{
    Int arr[] = { 2,3,4,10,40}
Int n = size of (arr) / size of (arr [0]);
 int x = 10;
 int result = binary search (arr, 0, n-1, x)
 (result == -1) ? Print 7 ("Element is not Pres")
                : Print 7 ("Element is
                  Present at index
                  result);
3 return 0;
```