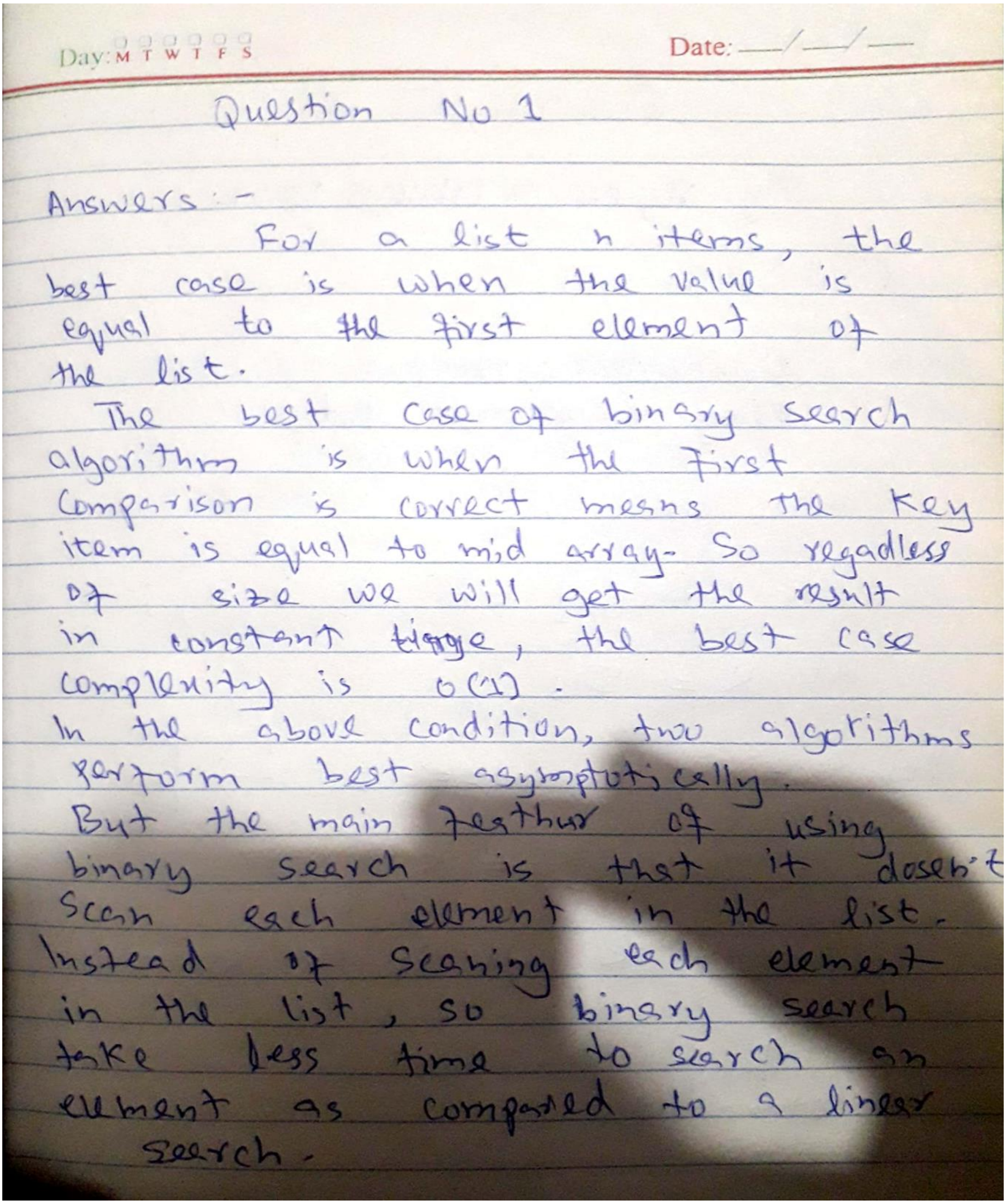
Name Muhammad Ali

Reg.no 19pwcse1801

Section A



Day: M T W T F S Date: __/__/__

Question No 1

Answers :-

For a list n items, the best case is when the value is equal to the first element of the list.

The best case of binary search algorithm is when the First Comparison is correct means the Key item is equal to mid array. So regadless of size we will get the result in constant time, the best case complexity is O(1).

In the above condition, two algorithms perform best asymptotically.

But the main feathur of using binary search is that it dosen't Scan each element in the list. Instead of Scaning each element in the list, so binary search take less time to search an element as compared to a linear search.

Day.

Question No

My reg.No = 19PWCSE 1801

Code :⇒

```
#include <iostream>
Using namespace std;

int main () {
int      arr[ ] = {1, 9, 8, 0};
int  K = 8;
for (int j = 4 ; j>=0 ; j--) {
 if (arr[j] == K) {
 cout << " K value is searched";
 }}
 return 0; }
```

Output = 8

Question NO 4

eliminate redundant element from name

Code:-

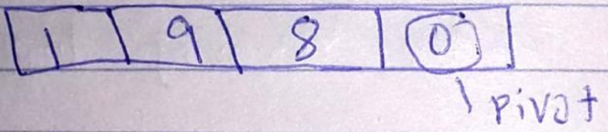
```
#includ <string.h>
int main() {
   Char  s [100] , temp=1, c=" ";
  int     i, j, k=0, n ;
 Prinf ("Enter string to redundant");
    gets (s);

 For (i=0;   si(i) ; i++) {
    if (! s [i] ==c)) {
      for(j=i+1; s{i] ;  j++)) {
      if (s [i] == s[i} )
           s[j] = c;   } } }
  For (i=0  ;  s{i] ;   i++){
     s[i] = s[i+k];
   if (s[i] ==c) { k++ ; j--
```

Question No 5

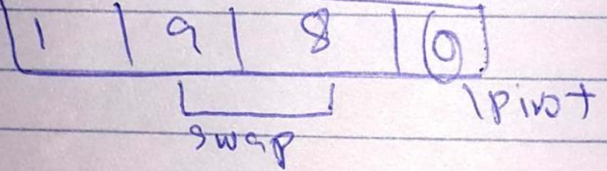
Reg. No = 19PWCSE 1801

array = [19 9, 8 , 0]

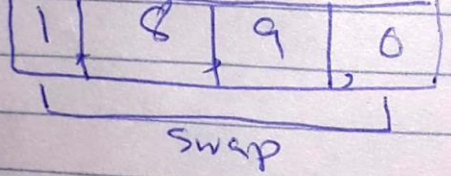
step 1 :-
Last inden number of array is pivot


1	9	8	0

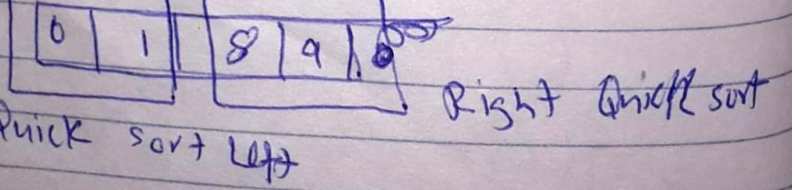
step 2 :- partition


1	9	8	0

step 3 :-


1	8	9	0

Sth :-


0	1	8	9	0

array = {0 1 , 8 , 9}

```
Printf (" string after removing
        all duplicate ");

Printf ("%s" , s) ;
  return 0;     }
```

Output ⇒ Enter string to redundent

```
Muhammad Ali
string after removing duplicate.
Muhamd Li
```

Qusstion No 6

Code :-

```
    #include <iostream>
  using namespace std;

class solution{
 puplic :
  string rD (string s, intk ) {

for  int j = 1 ; count=j ; j< s.size();
                                  j++) {
   count = s[i] = s[j - 1] ?
     count + 1 : 1 ;
  if (count ==k)

    s = rD(s.substr(0, j-k+1)+
              s.substr (j+1), k) ; }

     return s ;  }

int main () {
   int arr[] = {1, 9, 8, 6}
     int k = 4;
 string arrg = (arr{}, k);
 s. solution;
 return 0.
```

Question No7

Reg.No = 19Pwcse1801

1	9	8	0	5	2	7

Bubble sorting :-

First iteration :-

1	9	8	0	5	2	7

no chang

1	9	8	0	5	2	7

sort

1	8	9	0	5	2	7

sort

1	8	0	9	5	2	7

sort

1	8	0	5	9	2	7

sort

1	8	0	5	2	9	7

sort

1	8	0	5	2	7	9

2nd iteration :-

1	8	0	5	2	7	9

no change sort

1	0	8	5	2	7	9

sort

1	0	8	8	2	7	9

sort

1	0	5	2	8	7	9

sort

1	0	5	2	7	8	9

sort

3rd iteration :-

0	1	5	2	7	8	9

sort sort

0	1	2	5	7	8	9