DATE __/__/____
PAGE _____

Assignment 15

① int greatestElement(int a[], int n)
{
int i, d;
d = a[0];
for(i=1; i<n; i++)
{
if(a[i]>=d)
d = a[i];
}
return d;
}

② int smallestElement(int a[], int n)
{
int i, d;
d = a[0];
for(i=1; i<n; i++)
{
if(a[i]<=d)
d = a[i];
}
return d;
}

(3) void sort(int a[], int n)

{

int i, k, count, temp;

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

{

count = 0;

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

{

if(a[i]>=a[i+1])

{

temp = a[i];

a[i] = a[i+1];

a[i+1] = temp;

count = 1;

}

}

if(count == 0)

break;

}

(4) void rotateArray(int a[], int position , int d, int size)

```
{
    int i, temp, k;
if( d==1)
  {
  temp= a[0];
for(k=1; k<= position ; k++)
  {
  temp = a[0];
for(i=0; i<=(n-1) ; i++)
  a[i] = a[i+1];
  a[i] = temp;

  }
else
  {
for(k=1; k <= position ; k++)
    {
  temp = a[0];  temp = a[n-1];
for(i=0; i<=(n-1) ; i++)
                for( i= n-1 ; i>=1 ; i--)
  a[i] = a[i-1];
  a[i] = temp;

}
```

DATE: __/__/____
PAGE _____

(6) void displayReverse(int a[], int size)
{
int i;
for(i=n-1; i>=0; i--)
printf("%d ", a[i]);
}

(7) int countDuplicate(int a[], int size)
{
int i, j, k, count=0;
for(i=0; i<size; i++)
{
for(j=i-1; j>=0; j--)
{
if(a[j]==a[i])
break;
}
if(j>=0 && a[j]==a[i])
~~if(a[j]==a[i] && i>0)~~
continue;
for(k=i+1; k<size; k++)
{
if(a[k]==a[i])
{
count+=1;
break;
}
}
}
return count;

DATE __/__/____
PAGE _____

(8) Void print(Unique(int a[], int size)
{
int i, j, k, count;
for(i=0; i<size; i++)
{
count=0;
for(j=i-1; j>=0; j--)
{
if(a[j]==a[i])
break;
}
if(a[j]==a[i] if(j>=0 && a[j]==a[i])
continue;
for(k=i+1; k<n; k++)
{
if(a[k]==a[i])
{
count=4;
break;
}
}
if(count==0)
printf("%d ", a[i]);
}
}

DATE: __/__/____
PAGE _____

⑨
```
void sort(int a[], int b[], int size)
  {
int n, i, j, k=0, temp, d=0, count;
  n= 2*size;
int c[n];
for(i=0; i<size; i++)
  c[i] = a[i];
for(j=i; j<n; j++)
      {
  c[j] = b[k];
    k++;
  }

for(d=1; d<n; d++)
    {
  count = 0;
for(i=0; i<n; i++)
    {

if(a[i] <= a[i+1])
    {
  temp = a[i];
  a[i] = a[i+1];
  a[i+1] = temp;
    count = 1;
  }
}
if(count == 0)
  break;
}
for(i=0; i<n; i++)
  printf("%d ", c[i]);
}
```