1. What is the Output of Following Code?

```
void printArray(int *arr,int size)
{
      for (int i=0;i<size;++i)
            cout << arr[i] << ' ';

      cout << endl;
}

int main()
{
      int localArray[10] = {0,1,2,3,4,5,6,7,8,9};
      int *arr;

      arr = localArray + 3;
      printArray(arr,4);

      *arr = 15;
      arr[-2] = 7;
      printArray(localArray,10);

      unsigned char x = 0x01,y=0xff;
      unsigned char a = 203;
      x = x << 2;
      cout << (int) x << endl;

      x = a & y;
      cout << (int) x << endl;

      return 0;
}
```

OUTPUT:

2. What is the Output of Following Code?

```
void main()
{
      int i=0,j=0,total=4;
      char **strArr = new char *[total];
      int size = 7;
      for (i=0;i<total;++i)
      {
            strArr[i] = new char[size-i];
            strArr[i][size-i-1] = '\0';
      }
      for (i=0;i<total;++i)
      {     for (j=0;j<size-i-1;++j)
            {
                  strArr[i][j] = 'A' + j;
            }
      }
      for (i=0;i<total;++i)
            cout << strArr[i] << endl;

      for (i=0;i<total;++i)
            cout << (strArr[i]+1) << endl;

     cout << *strArr << endl;
     cout << **strArr << endl;
     cout << **strArr[3]<< endl;
     cout <<  strArr[2]+1 << endl;
     cout << *strArr[3]+2 << endl;
     cout << *(*strArr+2)+1 << endl;
     cout << *(strArr+2)[3] << endl;

}
```

OUTPUT:

3. **What is the Output of Following Code?**

```
int main() {
char *a[] = { "Argentina", "Korea", "Greece", "Nigeria"};
cout << *(a+1) << endl;
cout << *a[0] << endl;
cout << a[3] << endl;
cout << a[3][1] << endl;
return 0;
}
```

OUTPUT:

4. **What is the Output of Following Code?**

```
double *pt;
double a[3]={1.2, 2.3, 3.4};
pt=&a[1];
pt+=1;
```

```
cout<<*pt<<endl;
```

OUTPUT:

5.

What is the Output of Following Code?

```
int main(){
const char* what = "Is This";
what = "Interesting";
cout << *what;
what[3] = 'a';
cout << *what;
}
```

OUTPUT:

What is the Output of Following Code?

```
int& Now() {
int Where = 1;
return Where ;
}
int main() {
int Where;
Where= Now();
cout << Where;
}
```

OUTPUT:

How do the variables A and B differ?

```
char *const A = "Hi";
const char* B = "Hi";
```

Explain the problems with the following uses of C and D

```
const char* C = "hi mom";
C[3] = 'a';
char *const D = "hi mom";
D = "hi dad";
```

What is the Output of Following Code?

```
int x = 5;
int* y = new int(3);
int** z = &y;
int A[5] = {1,2,3,4,5};
cout << *y;
cout << **z;
cout << *&x;
cout << A[4];
cout << *(A+2);
cout << *(A+*y);
cout << A[**z];
cout << A[x];
```

Identify any error (dangling pointer, Memory Leak)

```
int * cube (int * a) {
      int s = *a**a**a;
      return &s;
}
int main(){
      int i =10;
      int j = *cube (&i);
      cout << j <<endl;
}
```

Identify any error (dangling pointer, Memory Leak)

```
int * square (int * a) {
int *s =new int;
```

Identify any error (dangling pointer, Memory Leak)

```
int meaning = 42;
int *life = &meaning;
```

```
 *s =  *a * *a;
return s;
}
int main(){
int i =10;
int j = * square (&i);
cout << j <<endl;
}
```

```
int **universe = &life;
int ***everything = &universe;
cout << ***everything <<endl;
delete life;
life = nullptr;
universe = nullptr;
everything = nullptr;
```

Identify any error (dangling pointer, Memory Leak)

```
void IncBy1( int * arr, int n){
  int * temp=new int[n+1];
  for(int i=0; i<n; i++)
      temp[i]=arr[i];
  delete [] arr;
  arr=temp;
}
//main
int * A = new int[5];
IncBy1(A, 5);
```

Identify any error (dangling pointer, Memory Leak)

```
int * product(int a, int b){
  int mul = a*b;
return & mul;
}
//main
int x = 7, y=10;
int * p =product(x,y);
cout<<*p;
```

Identify any error (dangling pointer, Memory Leak)

```
int a=5;
int * ptr = new int;
ptr[0]=a;
ptr=&a;
```

Identify any error (dangling pointer, Memory Leak)

```
char ** s = new char *[1];
char * name = new char[20];
strcpy(name,"John Doe");
s[0] = name;
delete [] name;
cout << s[0] << endl;
delete [] s;
s = nullptr;
```

Question:

Write a C++ program which takes four integer values from the user and rotate their values using a rotate function. For example the integer values are a=5, b=7, c=12, d=3 after rotation the values must be a=3, b=5, c=7 and d=12.

Note that you cannot use call by reference for this task

Quiz- 1 Given the following functions fill in the boxes bellow for main function.

```
void switchPtr (int *p,int *q)
{
      int *temp = p;
      p = q;
      q = temp;
}
```

```
int acceptPtr(int *p, int *q)
{
      *p = *q + 5;
      *q = *q + 10;
       p = q;
      *p = 5 + *q;
      *q = *p + 1;
      return *p+1;
}
```

```
int dontComplicate(int *p,int
*q,int &a,int &b)
{
      a = a+1;
      b = b+2;
      q = p;
      *p = 3;
      *q = 5;
      a = a+4;
      b = b+5;
      return *p + *q;
}
```

```
int notVerySimple(int *ptr1,int
*ptr2,int &a,int &b)
{
      *ptr1 = 1;
      *ptr2 = 5;
      a = 4;
      b = 3;
      return a+b;
}
```

Main Function	**x**	**y**	**Z**	**ptr1**	***ptr1**	**ptr2**	***ptr2**
`int main(){` `int x = 1, y = 2, z=0;`							
`int *ptr1 = &x;`							
`int *ptr2 = &y;`							
`switchPtr(ptr1,ptr2);`							
`z=acceptPtr(ptr1,ptr2);`							
`x = 0;     y = 1;`							
`z=dontComplicate(ptr2,ptr1,y,x);`							
`z=notVerySimple(ptr1,ptr2,x,y);`							
`return 0;` `}`							

From Sir Sarim's Home Work 1:

1) Find out the outputs of the following snippets of code by typing them into a complier.

   All of the following lines follow these declarations (only insert one snippet at a time, all are independent of each other):

```
int a=2,b=7,c=11;
int * aptr=&a, * bptr=&b, * cptr=&c;
int x[3]={5,9,11};
char y[6]={'H','E','L','L','O','\0'}, *sptr=NULL;
```

1	`cout<<&a<<" "<<&b<<" "<<&c;`
2	`cout<<*aptr<<" "<<*bptr<<endl;`

3	`*aptr=*bptr;`
	`cout<<a<<" "<<b;`

4	`bptr=cptr;`
	`cout<<*bptr<<endl;`

5	`aptr=bptr;`
	`bptr=cptr;`
	`cptr=aptr;`
	`cout<<*bptr<<" "<<*cptr<<endl;`

6	`cout<<x<<endl;`
7	`cout<<y<<endl;//compare with 6`
8	`cout<<(x+2)<<endl;//compare with 6`
9	`cout<<(y+2)<<endl;//compare with 7`
10	`cout<<*(x+2)<<" "<<*(y+2)<<endl;`
11	`cout<<x[2]<<" "<<y[2]<<endl;`
12	`cout<<&x[2]<<endl;//compare with 8`
13	`cout<<&y[2]<<endl;//compare with 9`
14	`cout<<1[x]<<" "<<1[y]<<endl;//why does this work?`
15	`cout<<*x+2<<" "<<*y+2<<endl;//compare with 10`

16	`aptr=x;`
	`cout<<*aptr<<endl;`

17	`aptr=x+1;`
	`cout<<*aptr<<endl;//compare with 16`

18	`sptr=y;`
	`cout<<*sptr<<endl;`

19	`sptr=y;`
	`sptr++;`
	`cout<<sptr<<endl;//compare with 18`

20	`(&a)[0]=-11;`
	`cout<<a<<endl;`

2) Write a function which prints every address of an integer array passed in parameter. The size of the array is also passed to the function.

3) Write a function that sorts an array of integers without using the subscript [ ] operator.