Question 1

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	int size,n,*arr;
	printf("enter the size of the array:");
	scanf("%d",&size);
	arr=(int *) malloc (size*sizeof(int));
	if(arr==NULL)
	{
		printf("error");
		return 0;
	}
	printf("enter the value of arrray:\n");
	for (int a=0;a<=size;a++)
	{
		scanf("%d",&n);
		arr[a]=n;

	}
	int length=sizeof(arr);
	printf("printed the element of repeation is:\n");


	for(int i=0;i<length;i++)
	{
		for(int j=i+1;j<length;j++)
		{
			if(arr[i]==arr[j])
			{
				printf("%d\n",arr[j]);
			}
		}
	}
	free(arr);
}
```

Output:

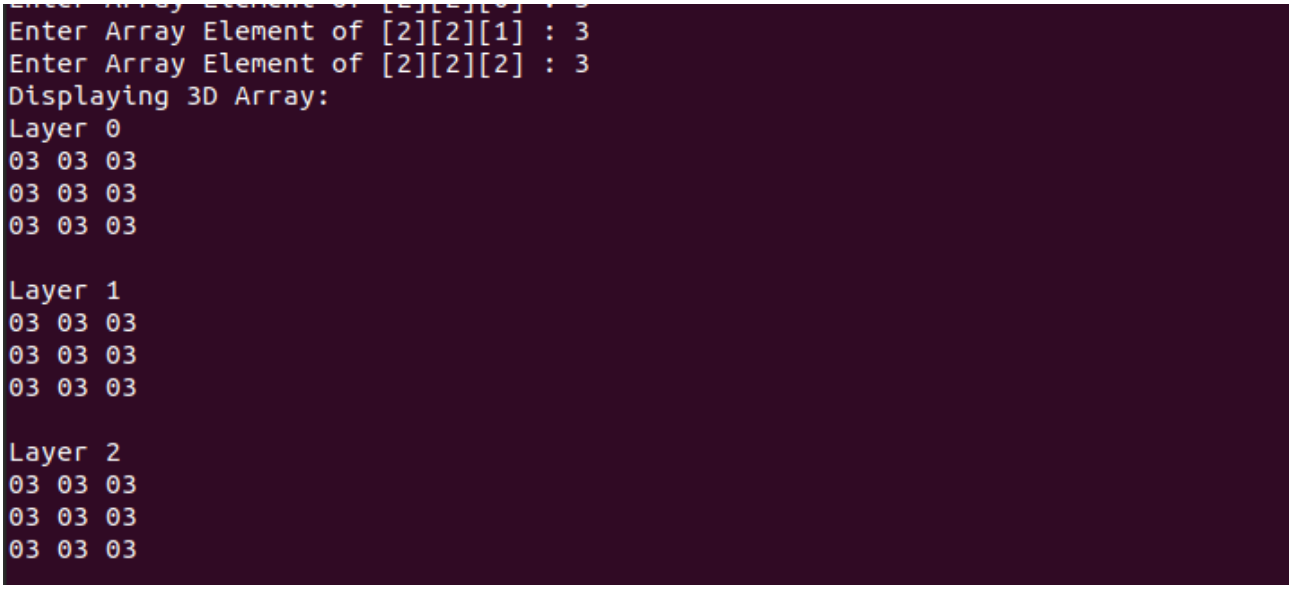
```
user@lab1d34-OptiPlex-3040:~/Desktop/Kezang Dorji/C Programming/lab 6$ ./questio
n1
enter the size of the array:2
enter the value of arrray:
3
4
4
printed the element of repeation is:
4
0
0
0
0
0
0
```

Question 2

```
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char* argv[])
{
  int rows = 0;
  int cols = 0;
  int height = 0;
  int ***array;
  int r, c, h;


  printf ("Enter 3D Array rows : ");
  scanf ("%d", &rows);

  printf ("Enter 3D Array columns : ");
  scanf ("%d", &cols);

  printf ("Enter 3D Array Layer : ");
  scanf ("%d", &height);

  array = (int ***) calloc (height,sizeof(int ***));
  for (h = 0; h < height; h++)
  {
    array[h] = (int **) calloc(rows,sizeof(int*));
    for (r = 0; r < rows; r++)
    {
      array[h][r] = (int *) calloc(cols,sizeof(int));
    }
  }

  for (h = 0; h < height; h++)
   {
    for (r = 0; r < rows; r++)
    {
      for (c = 0; c < cols; c++)
      {
        printf ("Enter Array Element of [%d][%d][%d] : ", h, r, c);
        scanf ("%d", &array[h][r][c]);
      }
    }
  }

  printf("Displaying 3D Array:\n");

  for (h = 0; h < height; h++)
  {
    printf("Layer %d\n", h);
    for (r = 0; r < rows; r++)
    {
```

```
      for (c = 0; c < cols; c++)
      {
        printf("%.2d ", array[h][r][c]);
      }
      printf("\n");
    }
    printf("\n");
  }

  return 0;
}
```

Output



Question 3

```
#include<stdio.h>
#include<math.h>
int product(int *i,int *j)
{
   int mul=*i**j;
   return mul;
}
int mean(int *i,int *j)
{
        int mid=(*i+*j)/2;
}
int deviation(int *i,int *j)
{
        int sum=*i+*j;
        int mean=sum/2;
        float SD=pow(sum-mean,2);
        return sqrt(SD/2);
}
int main()
{
```

```
        int a,b,mul;
        printf("give the value of a:");
        scanf("%d",&a);
        printf("give the value of b:");
        scanf("%d",&b);
        printf("the product is:%d\n",product(&a,&b));
        printf("the mean is:%d\n",mean(&a,&b));
        printf("the standard deviation is:%d",deviation(&a,&b));
        return 0;
}
```

Question 4

```
#include<stdio.h>
int main()

{
   int arr[] = {1, 2, 3, 4, 5};
   int length = sizeof(arr)/sizeof(arr[0]);
   int n = 2;
   printf("Before: \n");
   for (int i = 0; i < length; i++)
   {
      printf("%d ", arr[i]);
   }
   for(int i = 0; i < n; i++)
   {
      int j, first;
      first = arr[0];

      for(j = 0; j < length-1; j++)
      {
         arr[j] = arr[j+1];
      }
      arr[j] = first;
   }

   printf("\n");
   printf("After left rotation: \n");
   for(int i = 0; i < length; i++)
   {
      printf("%d ", arr[i]);
   }
   printf("\n");
   return 0;
}
```

Output

```
n4
Before:
1 2 3 4 5
After left rotation:
3 4 5 1 2
```

Question 5

```
#include <stdlib.h>
#include<stdio.h>
int main()
{

    // This pointer will hold the base address of the block created
    int* ptr;
    int n, i;

    // Get the number of elements for the array
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Dynamically allocate memory using malloc()
    ptr = (int*)malloc(n * sizeof(int));

    // Check if the memory has been successfully
    // allocated by malloc or not
    if (ptr == NULL) {
        printf("Memory not allocated.\n");
        exit(0);
    }
    else {

        // Memory has been successfully allocated
        printf("Memory successfully allocated\n");

        // Get the elements of the array
        printf("Enter the elements: \n");
        int a;
        for (i = 0; i < n; ++i) {
            scanf("%d", &a);
            ptr[i] = a;
        }

        // Print the elements of the array
        printf("The elements of the array are: ");
        for (i = 0; i < n; ++i) {
            printf("%d, ", ptr[i]);
        }
        printf("\n");
    }
```

```
    return 0;
}
```

Output

```
#include <stdlib.h>
#include<stdio.h>
int main()
{

    // This pointer will hold the base address of the block created
    int* ptr;
    int n, i;

    // Get the number of elements for the array
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Dynamically allocate memory using malloc()
    ptr = (int*)malloc(n * sizeof(int));

    // Check if the memory has been successfully
    // allocated by malloc or not
    if (ptr == NULL) {
        printf("Memory not allocated.\n");
        exit(0);
    }
    else {

        // Memory has been successfully allocated
        printf("Memory successfully allocated\n");

        // Get the elements of the array
        printf("Enter the elements: \n");
        int a;
        for (i = 0; i < n; ++i) {
            scanf("%d", &a);
            ptr[i] = a;
        }

        // Print the elements of the array
        printf("The elements of the array are: ");
        for (i = 0; i < n; ++i) {
            printf("%d, ", ptr[i]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
n5
Enter number of elements: 3
Memory successfully allocated
Enter the elements:
3
3
3
The elements of the array are: 3, 3, 3,
```