1. An array is a user-defined data type that stores related information together. Elements of array have the same data type. And array is needed for processing a large amount of data.
2. In the memory, array are stored in consecutive memory locations and are referenced by an index. This index make the user access the matrix.
3. A two-dimensional array is specified using two index where the first index denotes the row and the second denotes the column. And The C compiler treats two-dimensional array as a collection of an one-dimensional array.
4. Multi-dimensional array can contain as many indices as needed. So we can store the elements in 3d way.
5. Sparse matrix is a matrix that has large number of elements with a zero value. To efficiently utilize the memory, unique algorithms and structures should be used.
6. In two-dimensional arrays, the pointer which points to element is assigned a element address, the pointer which points to the one-dimensional array is assigned a row address. so we can write mat[i][j]=*(*(mat+i)+j)
7. Sparse matrix have a advantage of execution speed and amount of memory. Sparse data can be easily compressed, so it can reduce the memory usage. And there are two types of sparse matrices. One is all the elements above or below have zero value. This type of sparse matrixes is called lower triangular matrix and upper triangular matrix. Another is a non-zero value that it has non-zero value only on the diagonal or immediately above or below diagonal. This type of sparse matrix is called a tridiagonal matrix.
8. arr[35]=1000+2(35-0)=1070
9. addr(Marks[8][5])=2000+2{5(8-1)+(5-1)}=2078
10. The name of the array is the starting address of the array in memory. so we can use the name of array as a pointer to the first element of array. For example) Here is the arr[10], and we can use the name of array as a pointer(arr=&arr[0])
11. For example char*p[5] means an array of 5 pointers where each of the pointer points to an char variable.
12. There are two ways to pass a pointer to a function. The first is to pass each element as a parameter, and the second is to pass the entire array. Passing each element as a parameter can be divided into a case of giving a data value and a case of giving a data address.
13. add(Marks[8][5])=1000+2{10(5-1)+(8-1)}=1094
14. (a)

    i)If you need to move 7 spaces, Andrew will be inserted Name[0].
    ii)If you need to move 4 spaces, Andrew will be inserted Name[3].
    iii)If you need to move 5 spaces, Andrew will be inserted Name[2].
    iv)If you need to move 6 spaces, Andrew will be inserted Name[1].

(b)If you delete the name of esha, you must move the three names behind esha. And each of three names must be moved forward one space.

15. (a)The uninitialized portion of the array is filled with zeros.
(b)It does not compile.

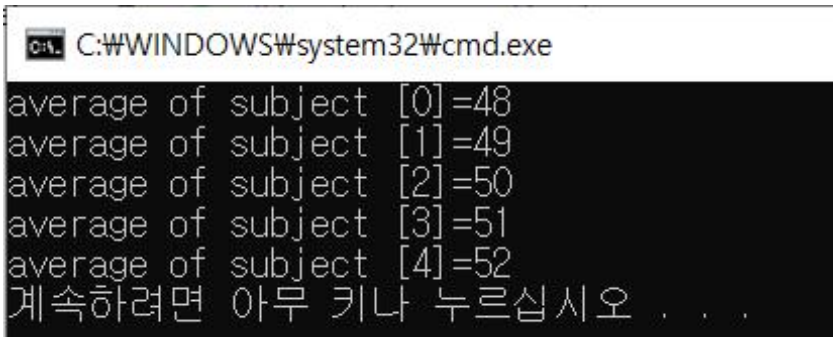
Programming Exercise

1.

(a)

```
#include <stdio.h>

int main()
{
        int marks[20][5]={
                                                {1,2,3,4,5},
                                                {6,7,8,9,10},
                                               {11,12,13,14,15},
                                               {16,17,18,19,20},
                                               {21,22,23,24,25},
                                               {26,27,28,29,30},
                                               {31,32,33,34,35},
                                               {36,37,38,39,40},
                                               {41,42,43,44,45},
                                               {46,47,48,49,50},
                                               {51,52,53,54,55},
                                               {56,57,58,59,60},
                                               {61,62,63,64,65},
                                               {66,67,68,69,70},
                                               {71,72,73,74,75},
                                               {76,77,78,79,80},
                                               {81,82,83,84,85},
                                               {86,87,88,89,90},
                                               {91,92,93,94,95},
                                               {96,97,98,99,100}
        };

        int sum=0;
        int avg,i,j;

        for(i=0;i<5;i++)
        {                       sum=0;
                                for(j=0;j<20;j++)
                                {
```

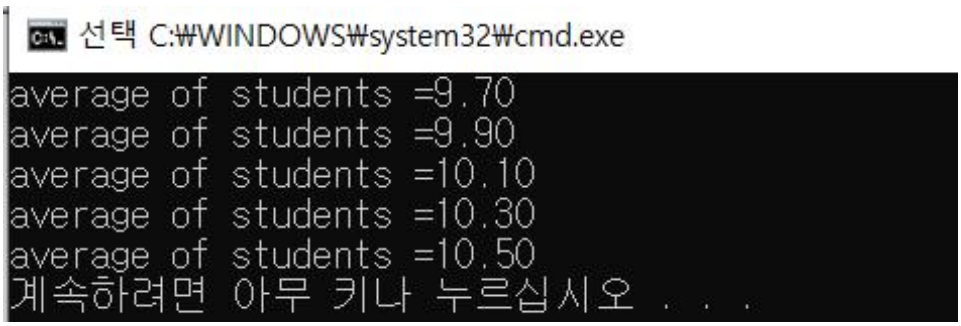
```
                    sum=sum+marks[j][i];
                }
                avg=sum/j;
                printf("average of subject [%d]=%d\n",i,avg);
        }

}
```



(b)

```
#include <stdio.h>

int main()
{
        int marks[20][5]={
                                {1,2,3,4,5},
                                {6,7,8,9,10},
                            {11,12,13,14,15},
                            {16,17,18,19,20},
                            {21,22,23,24,25},
                            {26,27,28,29,30},
                            {31,32,33,34,35},
                            {36,37,38,39,40},
                            {41,42,43,44,45},
                            {46,47,48,49,50},
                            {51,52,53,54,55},
                            {56,57,58,59,60},
                            {61,62,63,64,65},
                            {66,67,68,69,70},
                            {71,72,73,74,75},
                            {76,77,78,79,80},
                            {81,82,83,84,85},
                            {86,87,88,89,90},
                            {91,92,93,94,95},
                            {96,97,98,99,100}
        };
```

```
        int avg,i,j;
        int score;

        for(i=0;i<5;i++)
        {       score=0;
                for(j=0;j<20;j++)
                        {
                                score=score+marks[j][i];
                        }
                printf("average of students =%.2lf\n",score/100.0);
        }


}
```



(c)

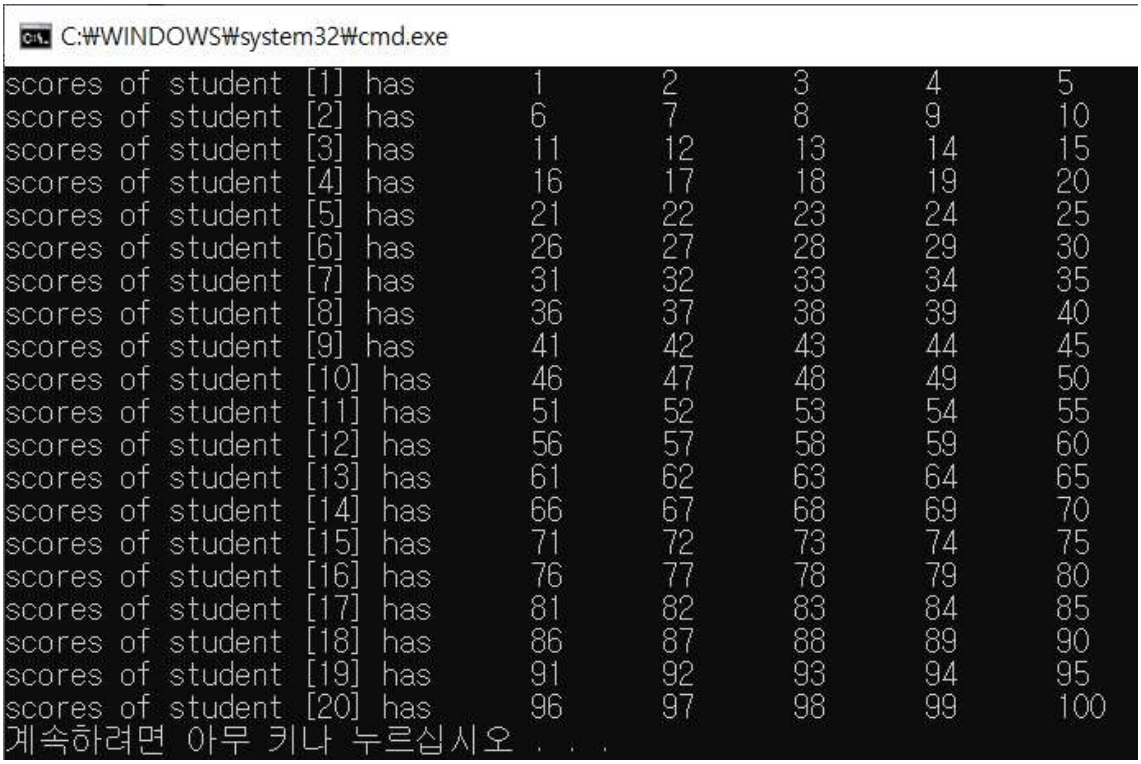
```
#include <stdio.h>

int main()
{
        int marks[20][5]={
                                                {1,2,3,4,5},
                                                {6,7,8,9,10},
                                        {11,12,13,14,15},
                                        {16,17,18,19,20},
                                        {21,22,23,24,25},
                                        {26,27,28,29,30},
                                        {31,32,33,34,35},
                                        {36,37,38,39,40},
                                        {41,42,43,44,45},
                                        {46,47,48,49,50},
                                        {51,52,53,54,55},
                                        {56,57,58,59,60},
                                        {61,62,63,64,65},
                                        {66,67,68,69,70},
```

```
                                        {71,72,73,74,75},
                                        {76,77,78,79,80},
                                        {81,82,83,84,85},
                                        {86,87,88,89,90},
                                        {91,92,93,94,95},
                                        {96,97,98,99,100}
        };


        int avg[20];
        int i,j;
        int score_sum=0;
        int count=0;
        for(i=0;i<20;i++)
        {               score_sum=0;
                        for(j=0;j<5;j++)
                        {
                                score_sum+=marks[i][j];

                        }
                        avg[i]=score_sum/j;

        }

        for(i=0;i<20;i++)
                if(avg[i]<50)
                        count++;

        printf("number of student who have scored below 50 is %d",count);
}
```

(d)

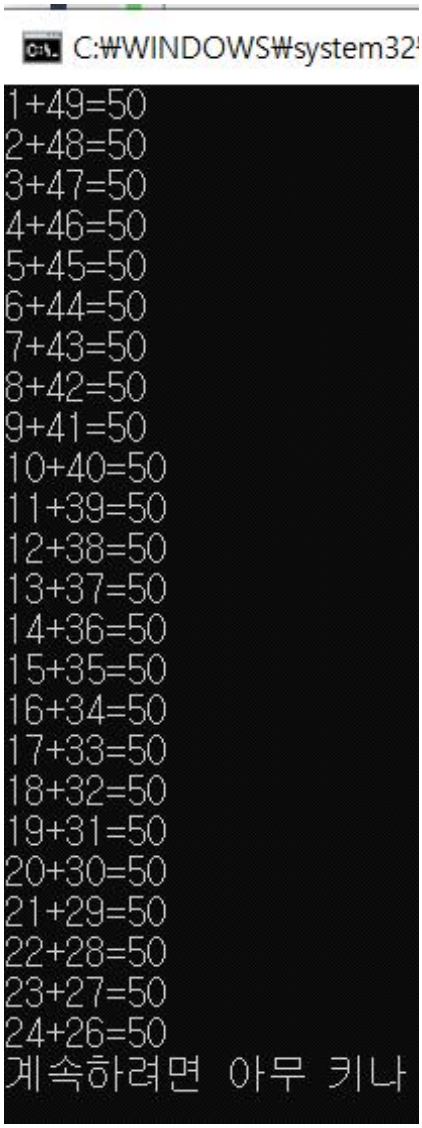
```
#include <stdio.h>

int main()
{
        int marks[20][5]={
                                                {1,2,3,4,5},
                                                {6,7,8,9,10},
                                               {11,12,13,14,15},
                                               {16,17,18,19,20},
                                               {21,22,23,24,25},
                                               {26,27,28,29,30},
                                               {31,32,33,34,35},
                                               {36,37,38,39,40},
                                               {41,42,43,44,45},
                                               {46,47,48,49,50},
                                               {51,52,53,54,55},
                                               {56,57,58,59,60},
                                               {61,62,63,64,65},
                                               {66,67,68,69,70},
                                               {71,72,73,74,75},
                                               {76,77,78,79,80},
                                               {81,82,83,84,85},
                                               {86,87,88,89,90},
                                               {91,92,93,94,95},
                                               {96,97,98,99,100}
        };


        int avg[20];
        int i,j;
        int score_sum=0;
        int count=0;
        for(i=0;i<20;i++)
        {                       printf("scores of student [%d] has",i+1);
                                for(j=0;j<5;j++)
                                {
                                  printf("\t%d",marks[i][j]);

                                }

                                printf("\n");
        }
```

}



2.

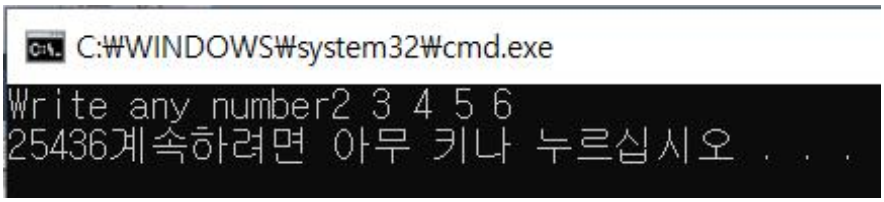
```
#include <stdio.h>

int main()
{
int a[100]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,
91,92,93,94,95,96,97,98,99,100};


        int i,j;
        for(i=0;i<99;i++)
                {
                        for(j=i+1;j<99;j++)
                                {
                                        if(a[i]+a[j]==50)
                                                printf("%d+%d=50\n",a[i],a[j]);
                                }
                }
}
```



3.

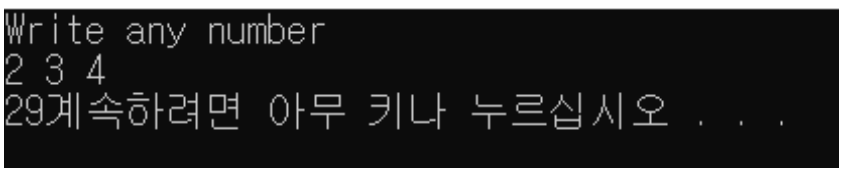
```
#include <stdio.h>


int main()
{
	int a[5];
	int i;
	int box;
	printf("Write any number");
	for(i=0;i<5;i++)
		scanf("%d",&a[i]);

	box=a[1];
	a[1]=a[(sizeof(a) / sizeof(int))-2];
```

```
	a[(sizeof(a) / sizeof(int))-2]=box;


	for(i=0;i<5;i++)
		printf("%d",a[i]);

}
```



4.

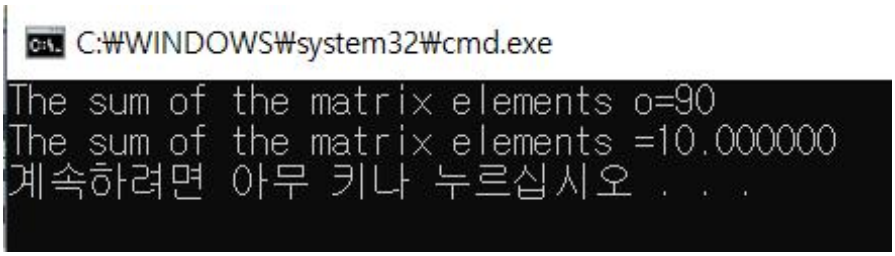
```
#include <stdio.h>


int main()
{
	int sum=0;
	int i;
	int a[5];

	printf("Write any number\n");
	for(i=0;i<5;i++)
		scanf("%d",&a[i]);



	for(i=0;i<5;i++)
		sum=sum+(a[i]*a[i]);

	printf("%d",sum);

}
```



5.

```
#include <stdio.h>

int main()
```

```
{
	int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};
	int b[3][3]={{1,2,3},{4,5,6},{7,8,9}};
	int c[3][3];
	int mean,sum=0;
	int total;
	int i,j;
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			c[i][j]=a[i][j]+b[i][j];
			sum=sum+c[i][j];

		}
	}
	printf("The sum of the matrix elements o=%d\n",sum);
	printf("The sum of the matrix elements =%lf\n",sum/9.0);


}
```



6.

```
#include <stdio.h>
int square(int num);
int main()
{
	int num;

	printf("write a any integer ");
	scanf("%d",&num);
	printf("%d\n",square(num));

}

int square(int num)
{
	return num*num;
```

```
}
```

```
write a any integer 23
529
계속하려면 아무 키나 누르십시오 . . .
```

7.

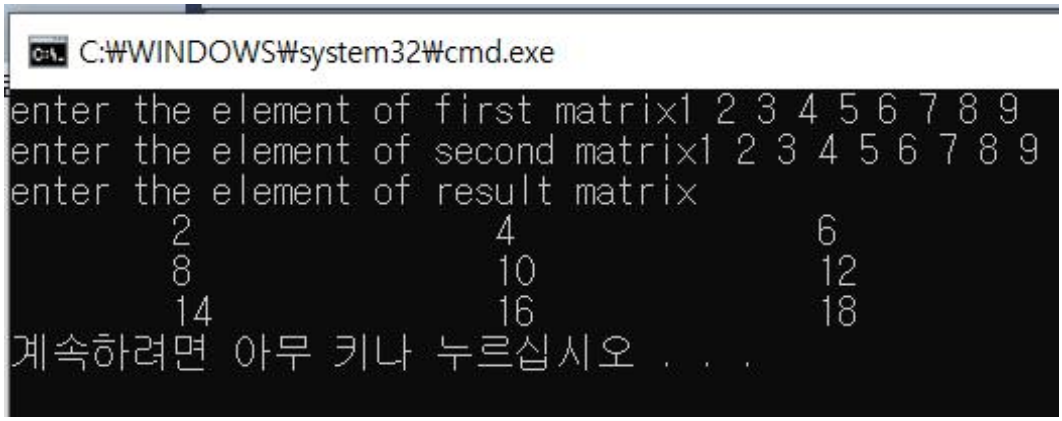
```
#include <stdio.h>
int sum_diagonal(int (*p)[3]);
int main()
{
	int (*p)[3];

	int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};
	p=a;
	printf("sum_diagonal=%d",sum_diagonal(a));

}

int sum_diagonal(int (*p)[3])
{
	int i;
	int sum=0;
	for(i=0;i<3;i++)
		sum=sum+p[i][i];
	return sum;
}
```

```
sum_diagonal=15계속하려면 아무 키나 누르십시오 . . .
```

8.

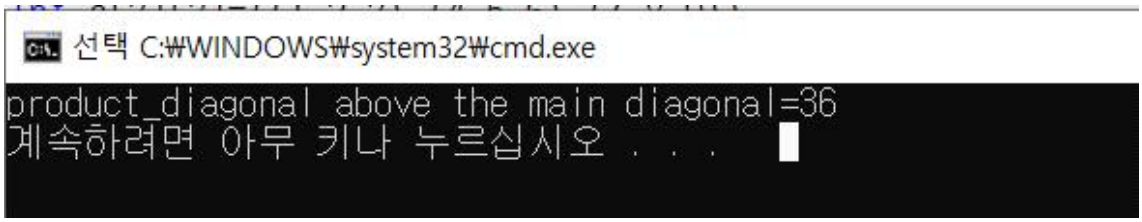
```
#include <stdio.h>

int main()
{
	int a[3][3],b[3][3],c[3][3];

	int i,j;

	printf("enter the element of first matrix");
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
```

```
		{
			scanf("%d",&a[i][j]);
		}
	}

	printf("enter the element of second matrix");
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			scanf("%d",&b[i][j]);
		}
	}

	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			*(*(c+i)+j)=*(*(a+i)+j)+*(*(b+i)+j);
		}
	}

	printf("enter the element of result matrix\n");
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			printf("\t%d\t",c[i][j]);
		}
		printf("\n");
	}


}
```



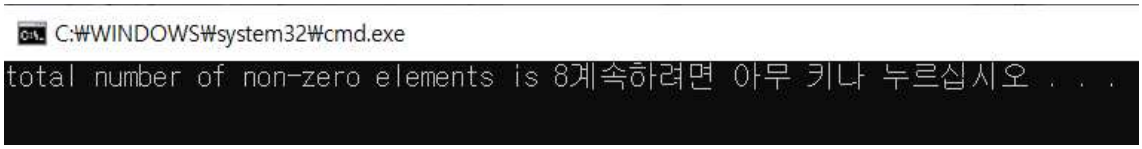
9.

```
#include <stdio.h>
int product_diagonal(int (*p)[3]);
int main()
{
	int (*p)[3];

	int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};
	p=a;
	printf("product_diagonal above the main diagonal=%d\n",product_diagonal(a));

}

int product_diagonal(int (*p)[3])
{
	int i,j;
	int product=1;
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				if(i<j)
				product=product*p[i][j];
			}
		}
	return product;
}
```



10.

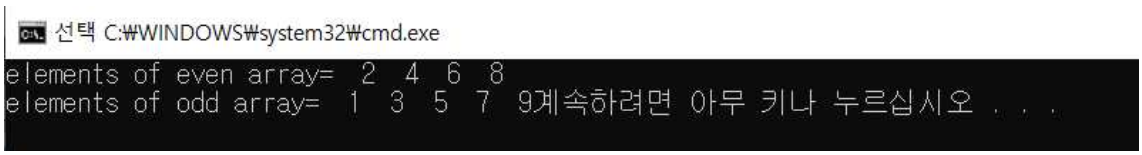
```
#include <stdio.h>

int main()
{
	int i,j;
	int sum=0;
	int count=0;
	int a[3][3]={{1,0,3},{4,5,6},{7,8,9}};

	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
				{
					if(a[i][j]!=0)count++;
				}
		}

	printf("total number of non-zero elements is %d",count);
}
```



11.

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

int main()
{
	int i,j;
	int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};
	int b[10]={0};//even
	int c[10]={0};//odd
	int index_1=0;
	int index_2=0;
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
				{
					if(a[i][j]%2==0)
						{
							b[index_1]=a[i][j];
```

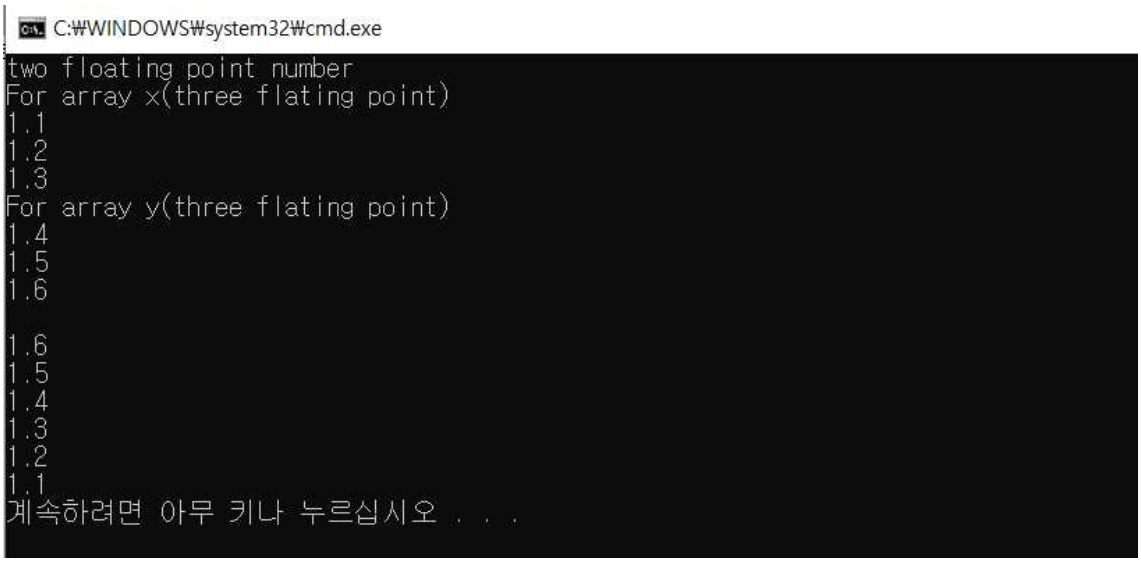
```
                                                index_1++;
                                        }
                                else
                                        {
                                                c[index_2]=a[i][j];
                                                index_2++;
                                        }
                                }
                        }

        printf("elements of even array=");
        for(i=0;i<10;i++)
                if(b[i]!=0)
                        printf("   %d",b[i]);
                printf("\n");

printf("elements of odd array=");
        for(i=0;i<10;i++)
                if(c[i]!=0)
                        printf("   %d",c[i]);
}
```



12.

```
#include <stdio.h>

int main()
{
        double x[3],y[3],z[6];
        int i;
        int index=0;
        printf("two floating point number\n");
        printf("For array x(three flating point)\n");
        for(i=0;i<3;i++)
                scanf("%lf",&x[i]);
        printf("For array y(three flating point)\n");
        for(i=0;i<3;i++)
                scanf("%lf",&y[i]);
```

```
        for(i=0;i<3;i++)
                {
                        z[index]=x[i];
                        index++;
                }
        for(i=0;i<3;i++)
                {
                        z[index]=y[i];
                        index++;
                }
        printf("\n");
        for(i=5;i>=0;i--)
                printf("%.1lf\n",z[i]);
}
```



13.

```
#include <stdio.h>
int find_secondlarge(int a[], int n);
int find_secondsmall(int a[], int n);
int main()
{

    int x, y;
        int i,j;
        int temp;
    int a[5] = { 7,1,2,3,4 };
    x=find_secondlarge(a, 5);
        y=find_secondsmall(a, 5);

        for(i=0;i<5;i++)
```

```
        {
                for(j=0;j<5;j++)
                {
                        if((a[i]==x)&&(a[j]==y))
                                {
                                        temp=a[i];
                                        a[i]=a[j];
                                        a[j]=temp;
                                }
                }
        }
        for(i=0;i<5;i++)
        {
                printf("%d",a[i]);
        }
}

int find_secondlarge(int a[], int n)
{
    int i;
        int large_index=-1;
    int large, second_large;
    int pos;

    second_large = large = 0;
    for (i = 0; i < n; i++)
    {


        if (a[i] > large)
        {
            second_large = large;
            pos = large_index;
            large = a[i];
            large_index = i;

        }
        else if (a[i] > second_large)
        {
            second_large = a[i];
            pos = i;

        }
```

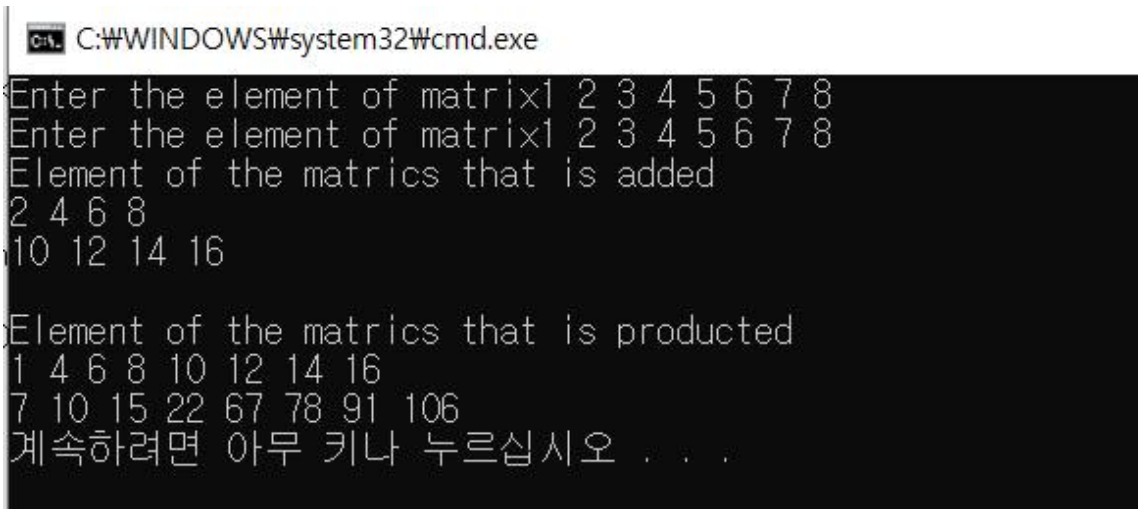
```
    }
    return pos;
}

int find_secondsmall(int a[], int n)
{
    int i;
    int small, second_small;
        int pos;
        int small_index=-1;
    second_small = small = a[0];


    for (i = 1; i < n; i++)
    {

        if (a[i] < small)
        {
            second_small = small;
                        pos=small_index;
            small = a[i];
                        small_index=i;
        }
        else if (a[i] < second_small)
        {
            second_small = a[i];
            pos = i;
        }

    }
    return pos;
}
```

14.

```
#include <stdio.h>
void read_matrix(int a[][2][2],int n);
void sum_matrix(int a[][2][2], int b[][2][2], int c[][2][2],int n);
void pro_matrix(int a[][2][2], int b[][2][2], int c[][2][2],int n);
int main()
{
```

```
        int a[2][2][2];
        int b[2][2][2];
        int c[2][2][2];
        int i,j,k;

        read_matrix(a,2);
        read_matrix(b,2);
        printf("Element of the matrics that is added\n");
        sum_matrix(a,b,c,2);
        printf("\n");
        printf("Element of the matrics that is producted\n");
        pro_matrix(a,b,c,2);

}

void read_matrix(int a[][2][2],int n)
{
        int i,j,k;
        printf("Enter the element of matrix");
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                        {
                                for(k=0;k<n;k++)
                                {
                                        scanf("%d",&a[i][j][k]);
                                }
                        }
        }
}

void sum_matrix(int a[][2][2], int b[][2][2], int c[][2][2],int n)
{
        int i,j,k;

        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                        {
                                for(k=0;k<n;k++)
                                {
                                        c[i][j][k]=a[i][j][k]+b[i][j][k];
                                        printf("%d ",c[i][j][k]);
```

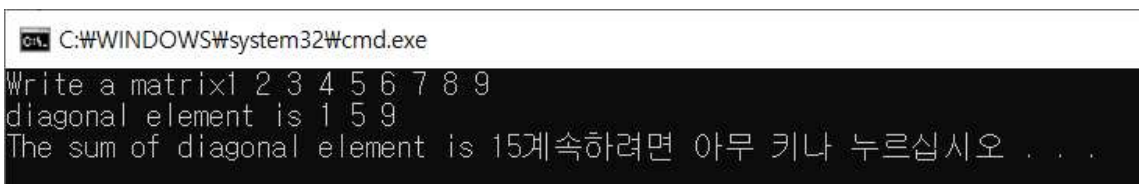
```
                                        }
                                }
                        printf("\n");
                }
}
void pro_matrix(int a[][2][2], int b[][2][2], int c[][2][2],int n)
{
        int i,j,k,r;

for(r=0;r<n;r++)
{
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                        {       c[r][i][j]=0;
                                for(k=0;k<n;k++)
                                {
                                        c[r][i][j]=c[r][i][j]+a[r][i][k]*b[r][k][j];
                                        printf("%d ",c[i][j][k]);
                                }
                        }

        }
        printf("\n");
}


}
```



15.

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

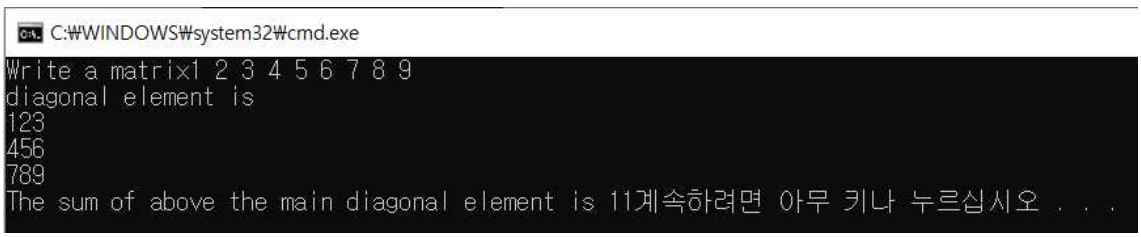
```
int main()
{
	int a[3][3];

	int i,j;
	int sum=0;
	printf("Write a matrix");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				scanf("%d",&a[i][j]);
			}

		}

	printf("diagonal element is");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				if(i==j)
					{
						printf(" %d",a[i][j]);
						sum+=a[i][j];
					}
			}
		}
	printf("\n");
	printf("The sum of diagonal element is %d",sum);
}
```



16.

```
#include <stdio.h>

int main()
{
	int a[3][3];
```

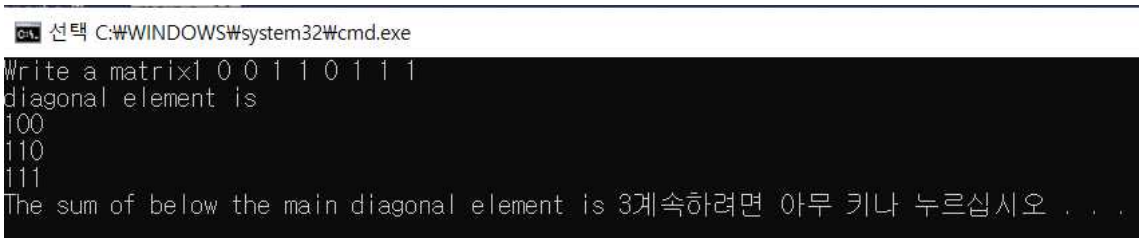
```
	int i,j;
	int sum=0;
	printf("Write a matrix");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				scanf("%d",&a[i][j]);
			}

		}

	printf("diagonal element is\n");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				if(i<j)
					{
						sum+=a[i][j];
					}
			}
		}
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			printf("%d",a[i][j]);
		}
		printf("\n");
	}
	printf("The sum of above the main diagonal element is %d",sum);
}
```



17.

```
#include <stdio.h>

int main()
```

```
{
	int a[3][3];

	int i,j;
	int sum=0;
	printf("Write a matrix");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				scanf("%d",&a[i][j]);
			}

		}

	printf("diagonal element is\n");
	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				if(j<i)
					{
						sum+=a[i][j];
					}
			}
		}
	for(i=0;i<3;i++)
	{
		for(j=0;j<3;j++)
		{
			printf("%d",a[i][j]);
		}
		printf("\n");
	}
	printf("The sum of below the main diagonal element is %d",sum);
}
```

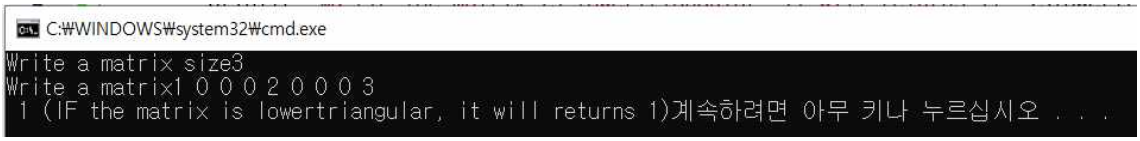


```
#include <stdio.h>
int isupperTriangular(int a[][5],int n);
int main()
{
	int a[5][5];

	int i,j,size;
	printf("Write a matrix size");
	scanf("%d",&size);
	printf("Write a matrix");
	for(i=0;i<size;i++)
		{
			for(j=0;j<size;j++)
			{
				scanf("%d",&a[i][j]);
			}

		}

		printf(" %d (IF the matrix is uppertriangular, it will returns 1)",isupperTriangular(a,size));
}

int isupperTriangular(int a[][5],int n)
{
	int i,j;
	int flag=0;
	for(i=0;i<n;i++)
		{
			for(j=0;j<n;j++)
			{
				if(i>j && a[i][j]!=0)
					{
						flag=1;
					}

			}
		}
	if(flag==1)return 0;
	 if(flag==0)return 1;
}
```

19.

```
#include <stdio.h>
int islowerTriangular(int a[][5],int n);
int main()
{
	int a[5][5];

	int i,j,size;
	printf("Write a matrix size");
	scanf("%d",&size);
	printf("Write a matrix");
	for(i=0;i<size;i++)
		{
			for(j=0;j<size;j++)
			{
				scanf("%d",&a[i][j]);
			}

		}

		printf(" %d (IF the matrix is lowertriangular, it will returns 1)",islowerTriangular(a,size));
}

int islowerTriangular(int a[][5],int n)
{
	int i,j;
	int flag=0;
	for(i=0;i<n;i++)
		{
			for(j=0;j<n;j++)
			{
				if(i<j && a[i][j]!=0)
					{
						flag=1;
					}

			}
		}
```

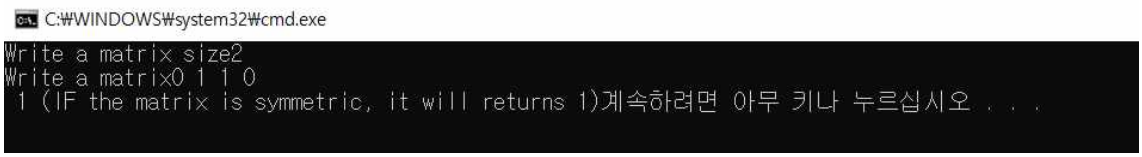
```
        if(flag==1)return 0;
    if(flag==0)return 1;
}
```



20.

```
#include <stdio.h>
int isSymmetric(int a[][5],int n);
int main()
{
        int a[5][5];

        int i,j,size;
        printf("Write a matrix size");
        scanf("%d",&size);
        printf("Write a matrix");
        for(i=0;i<size;i++)
                {
                        for(j=0;j<size;j++)
                        {
                                scanf("%d",&a[i][j]);
                        }

                }

                printf(" %d (IF the matrix is symmetric, it will returns 1)",isSymmetric(a,size));
}

int isSymmetric(int a[][5],int n)
{
        int i,j;
        int flag=0;

        for(i=0;i<n;i++)
                {
                        for(j=0;j<n;j++)
                        {
                                if(a[i][j]!=a[j][i]&&i!=j)
                                        {
                                                flag=1;
                                        }
```

```
                        }
                }

                 if(flag==1)return 0;
                if(flag==0)return 1;
}
```

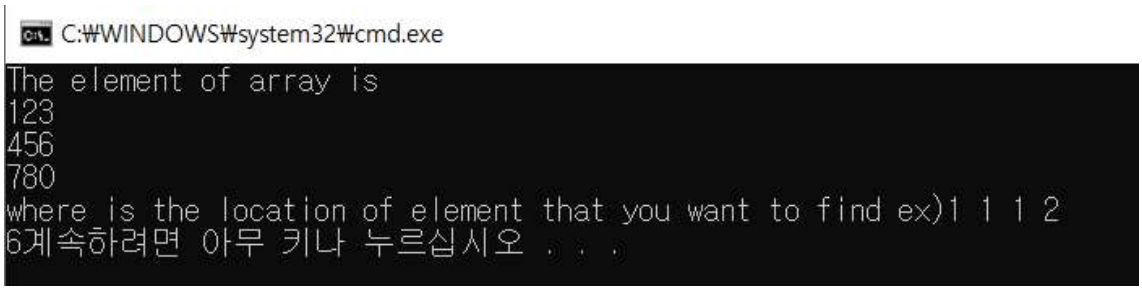


21.

```
#include <stdio.h>

int main()
{
        int A[2][2]={{1,2},{1,3}};
        int B[2][2]={{2,3},{3,4}};
        int C[2][2];
        int i,j;

        for(i=0;i<2;i++)
        {
                for(j=0;j<2;j++)
                {
                        A[i][j]=2*A[i][j];
                        B[i][j]=3*B[i][j];
                        C[i][j]=A[i][j]+B[i][j];
                }
        }
        for(i=0;i<2;i++)
        {
                for(j=0;j<2;j++)
                        {
                                printf(" %d",C[i][j]);
                        }
                printf("\n");
        }
}
```

22.

```
#include <stdio.h>

int main()
{
	int a[3][3]={{1,2,3},{4,5,6},{7,8.9}};
	int i,j,num1,num2;
	printf("The element of array is\n");

	for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
				printf("%d",a[i][j]);
			}
				printf("\n");
		}
	printf("where is the location of element that you want to find ex)1 1 ");
	scanf("%d %d",&num1,&num2);


	printf("%d",*(*(a+num1)+num2));
}
```



23.

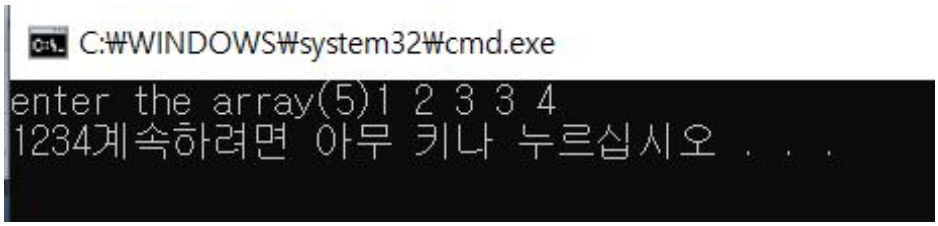
```
#include <stdio.h>

int main()

{
```

```
        int num;
        int digit[5]={0};
        int i=0,j;
        printf("Enter the number");
        scanf("%d",&num);
        while (num > 0)
{
    digit[i] = num%10;
    num /= 10;
        i++;

}
        j=i-1;
        while(digit[j]!=0)
        {
                if(j<0)
                        break;
                printf("%d ",digit[j]);
                j--;

        }
}
```

24.

```
#include <stdio.h>

int main()
{
        int a[5]={1,2,3,3,4};
        int b[5]={0};
        int flag=0;
        int i,j,k=0;
        b[0]=a[0];

        for(i=0;i<4;i++)
         {      for(j=i+1;j<5;j++)
                {
                        if(b[i]!=a[j])
                                {
```

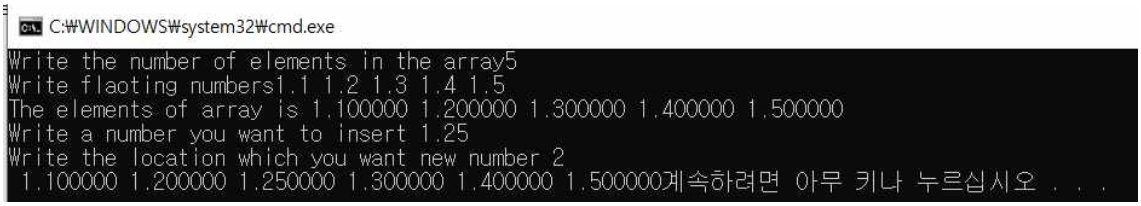
```
                                        b[i+1]=a[j];
                                        if(j==4)
                                                flag=1;
                                        break;
                                }

                }
                        if(flag==1)
                                break;
         }
        while(b[k]!=0)
        {
                printf("%d",b[k]);
                k++;
        }
}
```



25.

```
#include <stdio.h>

int main()
{
        double x[10];
        int i,location,n;
        double num;
        printf("Write the number of elements in the array");
        scanf("%d",&n);
        printf("Write flaoting numbers");
        for(i=0;i<n;i++)
        scanf("%lf",&x[i]);
        printf("The elements of array is");
        for(i=0;i<n;i++)
        printf(" %lf",x[i]);
        printf("\n");
        printf("Write a number you want to insert");
        scanf("%lf",&num);
        printf("Write the location which you want new number");
        scanf("%d",&location);
```

```
        for(i=n-1;i>=location;i--)
                x[i+1]=x[i];
        x[location]=num;
        n=n+1;

        for(i=0;i<n;i++)
        printf("  %lf",x[i]);


}
```



Multiple-choice Questions

1. (b)
2. (d)
3. (d)
4. (b)
5. (b)
6. (d)

True or False

1. F
2. T
3. T
4. F
5. F
6. T
7. T
8. T
9. F
10. T
11. T
12. F
13. F
14. F
15. T

Fill in the Blancks

1. []
2. consecutive
3. n
4. starting address of the array
5. data type, name, size
6. Name of the array
7. elements
8. array of array
9. integeral
10. fourth