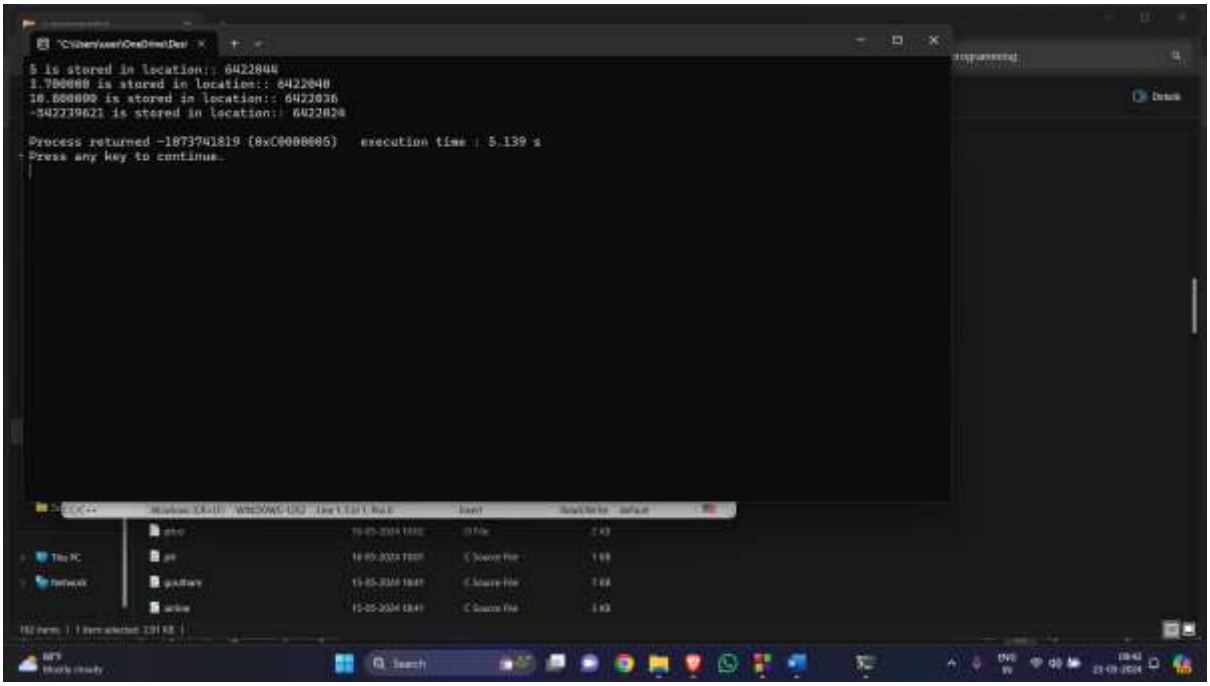
DAY 5

1. Pointer

```
#include<stdio.h>
#include<conio.h>
main()
{
   int a;
   float b,c;
   double d;
   char ch;
   system ("cls");
   a=5;b=3.7;c=10.80;d=12345678.99;ch='s';
   printf("%d is stored in location:: %u \n",a,&a);
   printf("%f is stored in location:: %u \n",b,&b);
   printf("%f is stored in location:: %u \n",c,&c);
   printf("%ld is stored in location:: %u \n",d,&d);
   printf("%s is stored in location:: %u \n",ch,&ch);



}
```

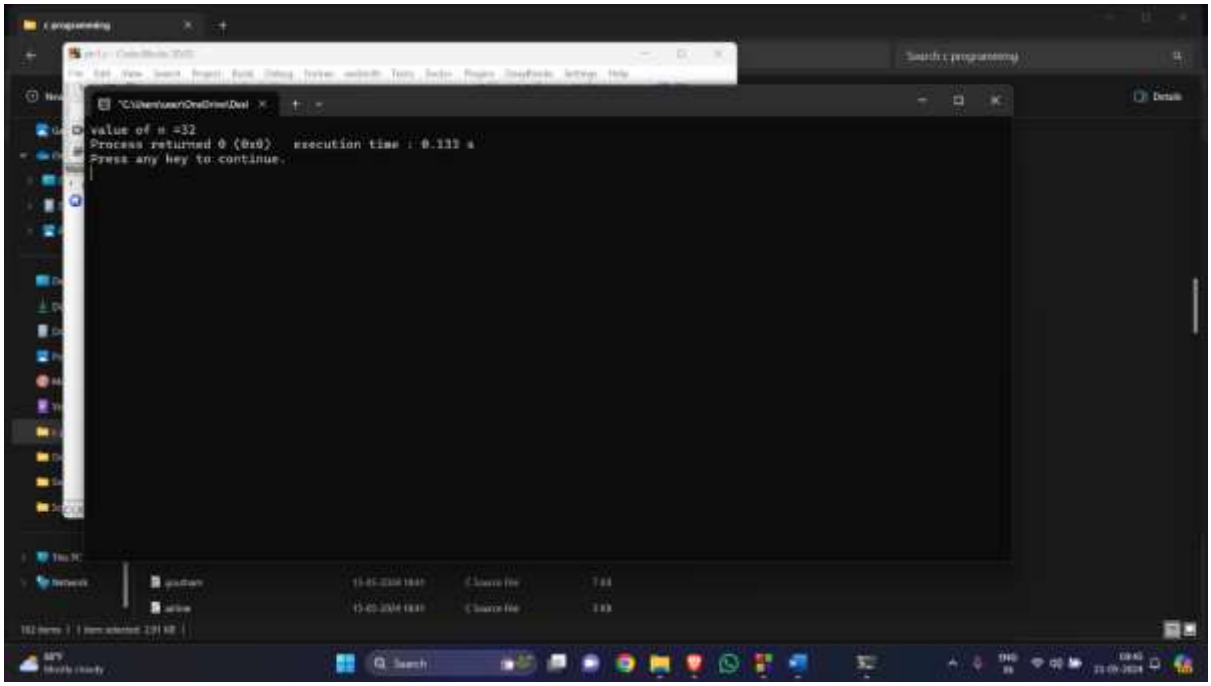
OUTPUT:



2. Pointer

```
#include<stdio.h>
main()
{
   int *p,n;
   p=&n;
   n=0x18;
   *p=*p+4;
   n=*p+4;
   printf("value of n =%d",n);
}
```

OUTPUT:

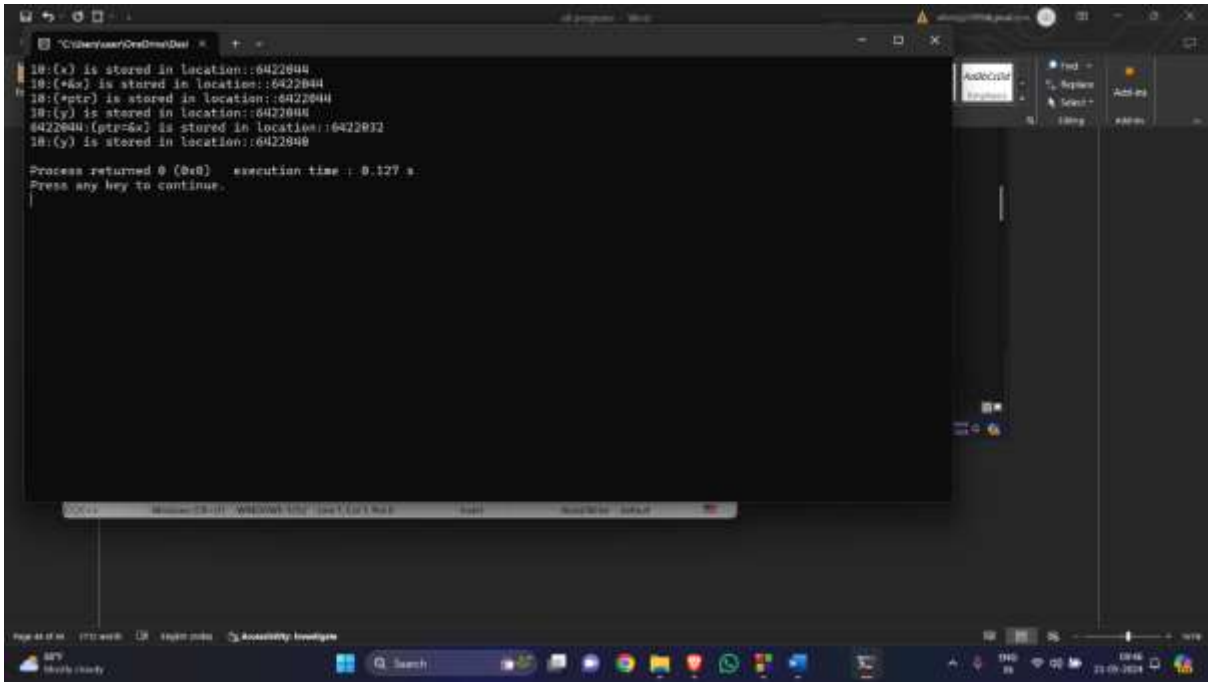


3. Pointer

```
#include<stdio.h>
main()
   {
      int x,y;
      int*ptr;
      x=10;
      ptr = &x;
      y= *ptr;
//       x, &x,
//       *&x, &x,
//       *ptr, ptr
//       , &*ptr
//       ptr, &ptr
//       y, &y
      printf("%d:(x) is stored in location::%u\n",x,&x);
      printf("%d:(*&x) is stored in location::%u\n",*&x,&x);
      printf("%d:(*ptr) is stored in location::%u\n",*ptr,ptr);
      printf("%d:(y) is stored in location::%u\n",y,&*ptr);
```

```
    printf("%u:(ptr=&x) is stored in location::%u\n",ptr,&ptr);
    printf("%d:(y) is stored in location::%u\n",y,&y);


}
```

OUTPUT:

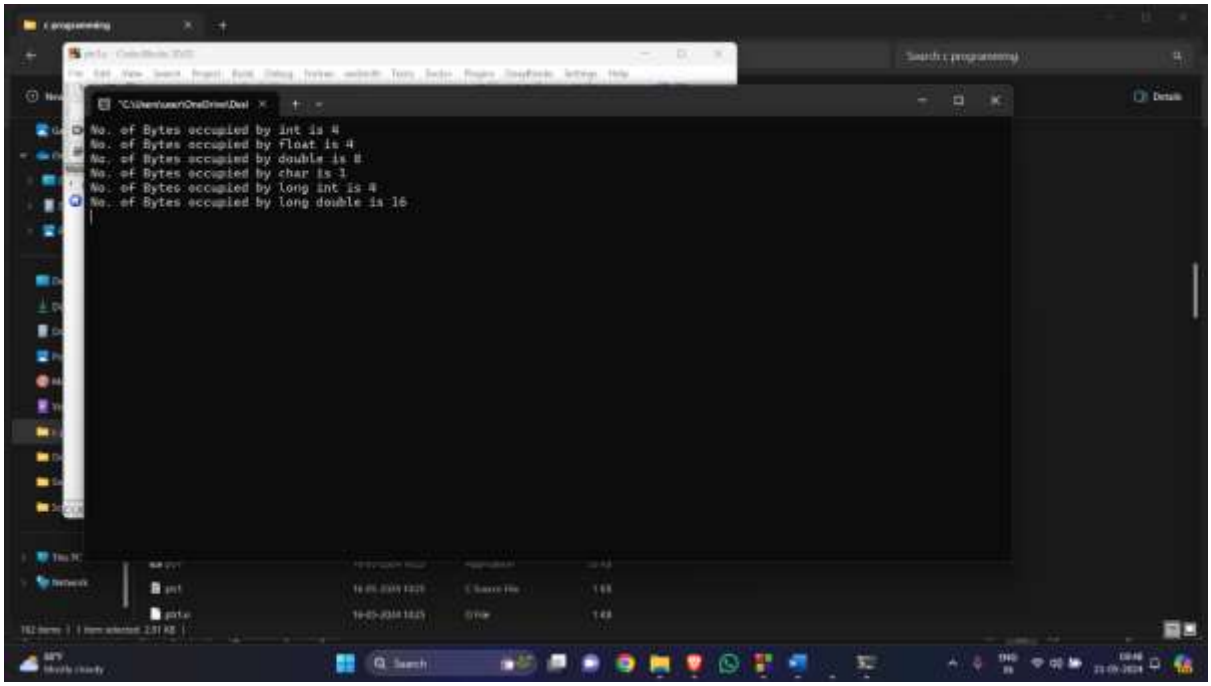


4. Size of data types

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

main()
{
  printf ("No. of Bytes occupied by int is %d \n", sizeof(int));
  printf ("No. of Bytes occupied by float is %d \n", sizeof(float));
  printf ("No. of Bytes occupied by double is %d \n", sizeof(double));
  printf ("No. of Bytes occupied by char is %d \n", sizeof(char));
  printf ("No. of Bytes occupied by long int is %d \n", sizeof(long int));
  printf ("No. of Bytes occupied by long double is %d \n", sizeof(long double));
  getch();
```

}

OUTPUT:

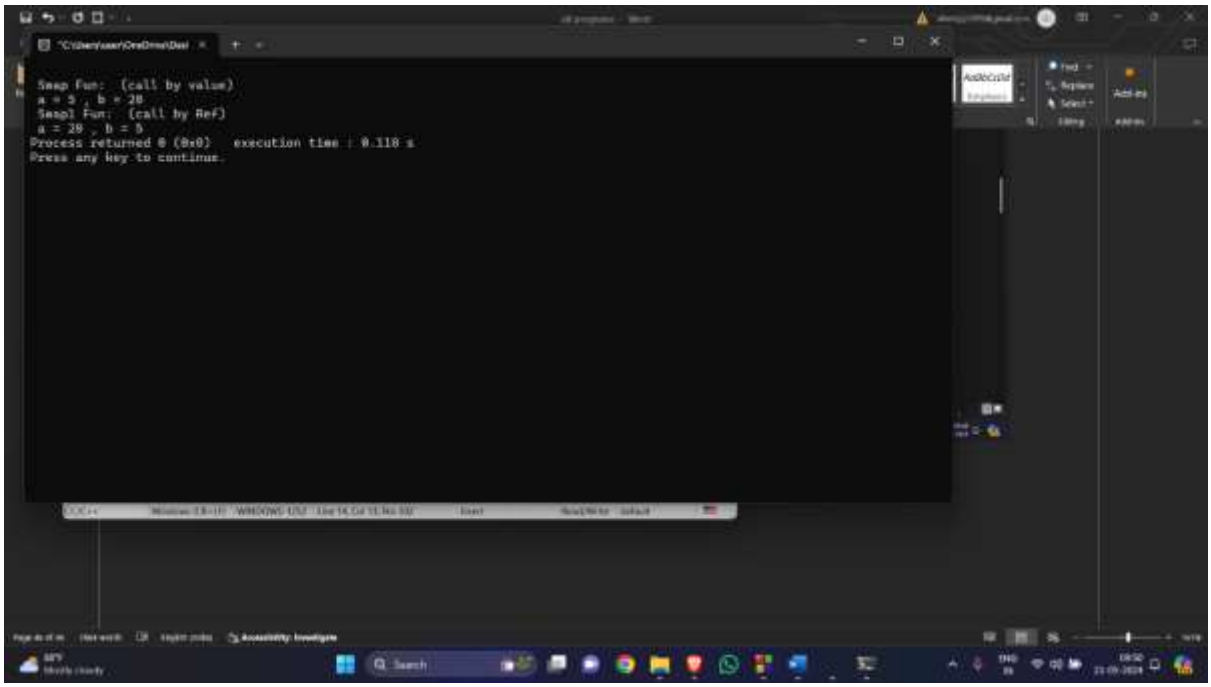


5. Call by reference

```
#include <stdio.h>
main()
{
   int a,b;
   a=5, b=20;
   swap (a,b);
   printf ("\n Swap Fun:  (call by value)  \n a = %d , b = %d ", a,b);
   swap1 (&a, &b);
   printf ("\n Swap1 Fun:  (call by Ref)  \n a = %d , b = %d ", a,b);
}
void swap (int x, int y)
{
   int tmp;
   tmp = x;
   x=y;
   y=tmp;
}
```

```
void swap1 (int *x1, int *y1)
{
   int tmp1;
   tmp1 = *x1;
   *x1=*y1;
   *y1=tmp1;
}
```

OUTPUT:

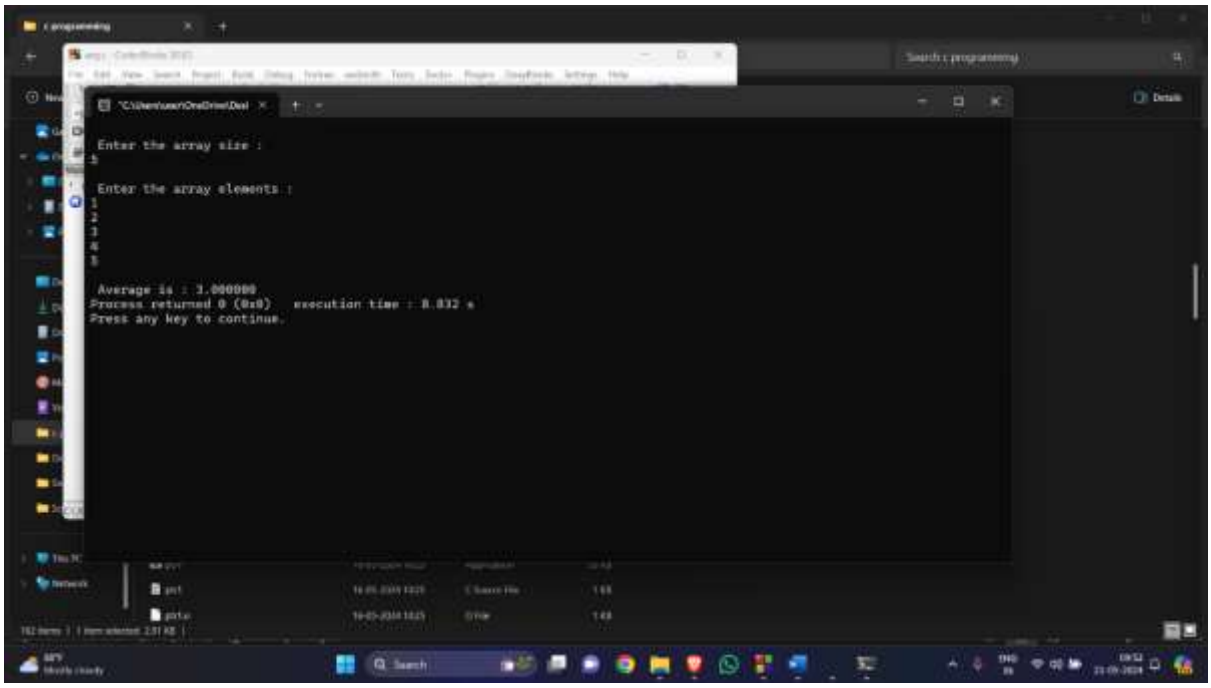


6. Average of a array

```
#include <stdio.h>
float avg (int arr[], int size);
main ()
{
   int x[100], k, n;
   printf("\n Enter the array size :\n");
   scanf ("%d",&n);
   printf("\n Enter the array elements :\n");
   for (k=0;k<n;k++)
   {
```

```
      scanf("%d", &x[k]);
     }
     printf("\n Average is : %f", avg (x,n));
}
float avg (int arr[], int size)
{
   int *p,i,sum=0;
   p=arr;
   for (i=0;i<size;i++)
   {
      sum = sum + *(p+i);
   }
   return (float) sum/size;
}
```

OUTPUT:

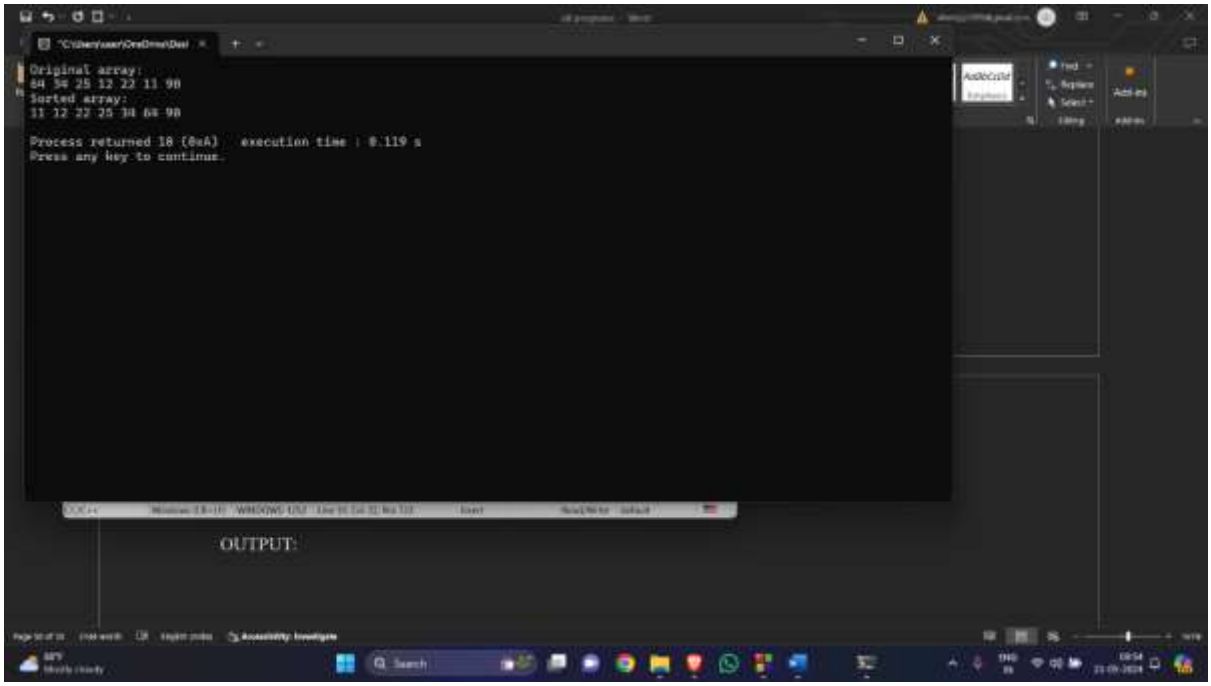


7. Bubble sort

```
#include <stdio.h>

void bubbleSort(int *arr, int n) {
```

```
    int i, j,temp;
    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (*(arr + j) > *(arr + j + 1)) {
                temp = *(arr + j);
                *(arr + j) = *(arr + j + 1);
                *(arr + j + 1) = temp;
            }
        }
    }
}

void main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
     printf("Original array: \n");
      for (int i = 0; i < n; i++)
    {
        printf("%d ", *(arr + i));
    }
    printf("\n");
    bubbleSort(arr, n);
    printf("Sorted array: \n");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", *(arr + i));
    }
    printf("\n");

}
```

OUTPUT:


Original array:
64 34 25 12 22 11 90
Sorted array:
11 12 22 25 34 64 90
Process returned 18 (0xA) execution time : 0.119 s
Press any key to continue.

OUTPUT: