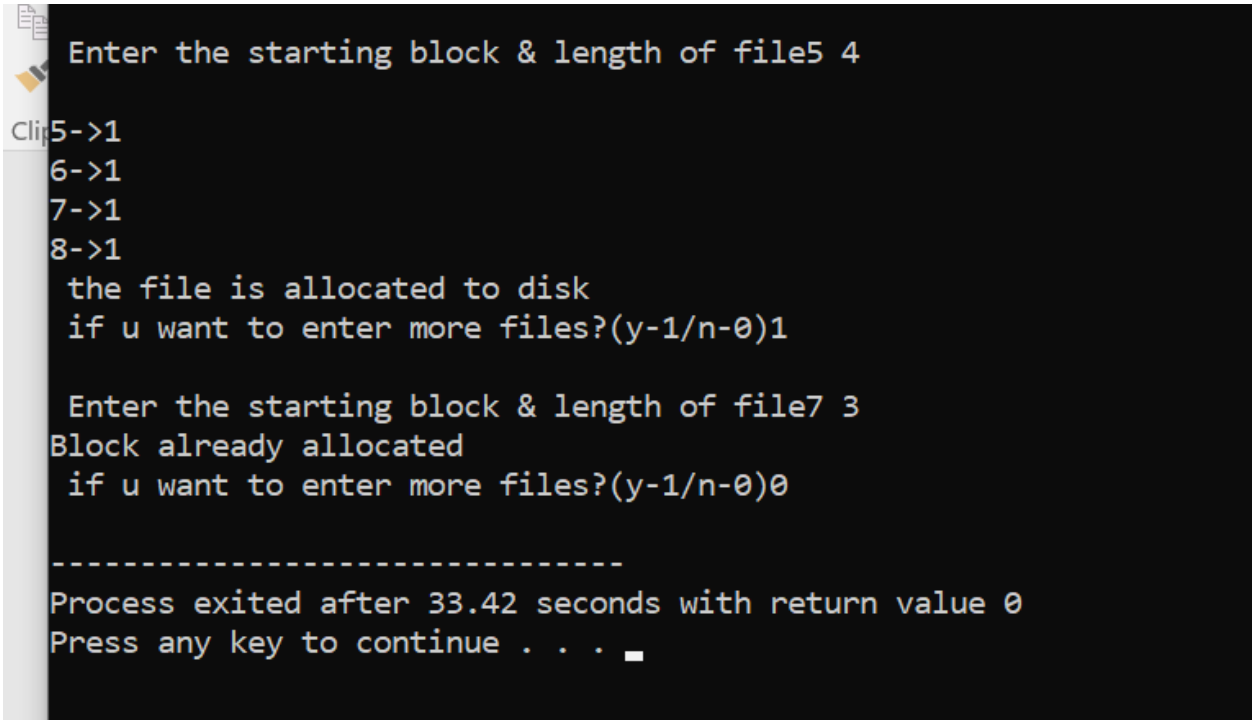
LAB # 14

Q) Implement the above code and paste the screen shot of the output.

a) SEQUENTIAL:

```
#include<stdio.h>
#include<conio.h>
main()
{ int f[50],i,st,j,len,c,k;
for(i=0;i<50;i++)
f[i]=0; X:
printf("\n Enter the starting block & length of file");
scanf("%d%d",&st,&len);
for(j=st;j<(st+len);j++)
if(f[j]==0)
{
f[j]=1;
printf("\n%d->%d",j,f[j]);
}
else
{
printf("Block already allocated");
break;
}
if(j==(st+len))
printf("\n the file is allocated to disk");
printf("\n if u want to enter more files?(y-1/n-0)");
scanf("%d",&c); if(c==1) goto X;
else
exit(0);
getch();
}
```



b) ***INDEXED:***

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

int main() {
   int f[50], i, k, j, inde[50], n, c, p;

   // Initialize all blocks to 0 (free)
   for(i = 0; i < 50; i++)
      f[i] = 0;

   while (1) {
      printf("\nEnter index block: ");
      scanf("%d", &p);

      if (p < 0 || p >= 50) {
         printf("Invalid block index! Try again.\n");
         continue;
      }

      if(f[p] == 0) {
```

```
            f[p] = 1;
            printf("Enter number of files on index: ");
            scanf("%d", &n);

            if (n > 50) {
                printf("Too many files! Try again.\n");
                f[p] = 0;
                continue;
            }

            printf("Enter the block numbers:\n");
            int valid = 1;
            for(i = 0; i < n; i++) {
                scanf("%d", &inde[i]);
                if(inde[i] < 0 || inde[i] >= 50 || f[inde[i]] == 1) {
                    valid = 0;
                }
            }

            if (!valid) {
                printf("One or more blocks are already allocated or invalid. Try
again.\n");
                f[p] = 0;
                continue;
            }

            for(j = 0; j < n; j++)
                f[inde[j]] = 1;

            printf("\nFile allocated successfully.\nIndexed block: %d\n", p);
            for(k = 0; k < n; k++)
                printf("%d -> %d : Allocated\n", p, inde[k]);
        }
        else {
            printf("Block already allocated. Try a different index.\n");
            continue;
        }

        printf("Enter 1 to enter more files, or 0 to exit: ");
        scanf("%d", &c);
        if (c != 1)
            break;
    }
```

```
    return 0;
}
```

```
Enter index block: 10
Enter number of files on index: 3
Enter the block numbers:
21
22
23

File allocated successfully.
Indexed block: 10
10 -> 21 : Allocated
10 -> 22 : Allocated
10 -> 23 : Allocated
Enter 1 to enter more files, or 0 to exit: 0

--------------------------------
Process exited after 41.33 seconds with return value 0
Press any key to continue . . .
```

c) ***LINKED:***

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

int main() {
    int f[50], i, k, j, inde[50], n, c, p;

    // Initialize all blocks to 0 (free)
    for(i = 0; i < 50; i++)
        f[i] = 0;

    while (1) {
        printf("\nEnter index block: ");
        scanf("%d", &p);
```

```
        if (p < 0 || p >= 50) {
            printf("Invalid block index! Try again.\n");
            continue;
        }

        if(f[p] == 0) {
            f[p] = 1;
            printf("Enter number of files on index: ");
            scanf("%d", &n);

            if (n > 50) {
                printf("Too many files! Try again.\n");
                f[p] = 0;
                continue;
            }

            printf("Enter the block numbers:\n");
            int valid = 1;
            for(i = 0; i < n; i++) {
                scanf("%d", &inde[i]);
                if(inde[i] < 0 || inde[i] >= 50 || f[inde[i]] == 1) {
                    valid = 0;
                }
            }

            if (!valid) {
                printf("One or more blocks are already allocated or invalid. Try
again.\n");
                f[p] = 0;
                continue;
            }

            for(j = 0; j < n; j++)
                f[inde[j]] = 1;

            printf("\nFile allocated successfully.\nIndexed block: %d\n", p);
            for(k = 0; k < n; k++)
                printf("%d -> %d : Allocated\n", p, inde[k]);
        }
        else {
            printf("Block already allocated. Try a different index.\n");
            continue;
```

```
        }

        printf("Enter 1 to enter more files, or 0 to exit: ");
        scanf("%d", &c);
        if (c != 1)
            break;
    }

    return 0;
}
```