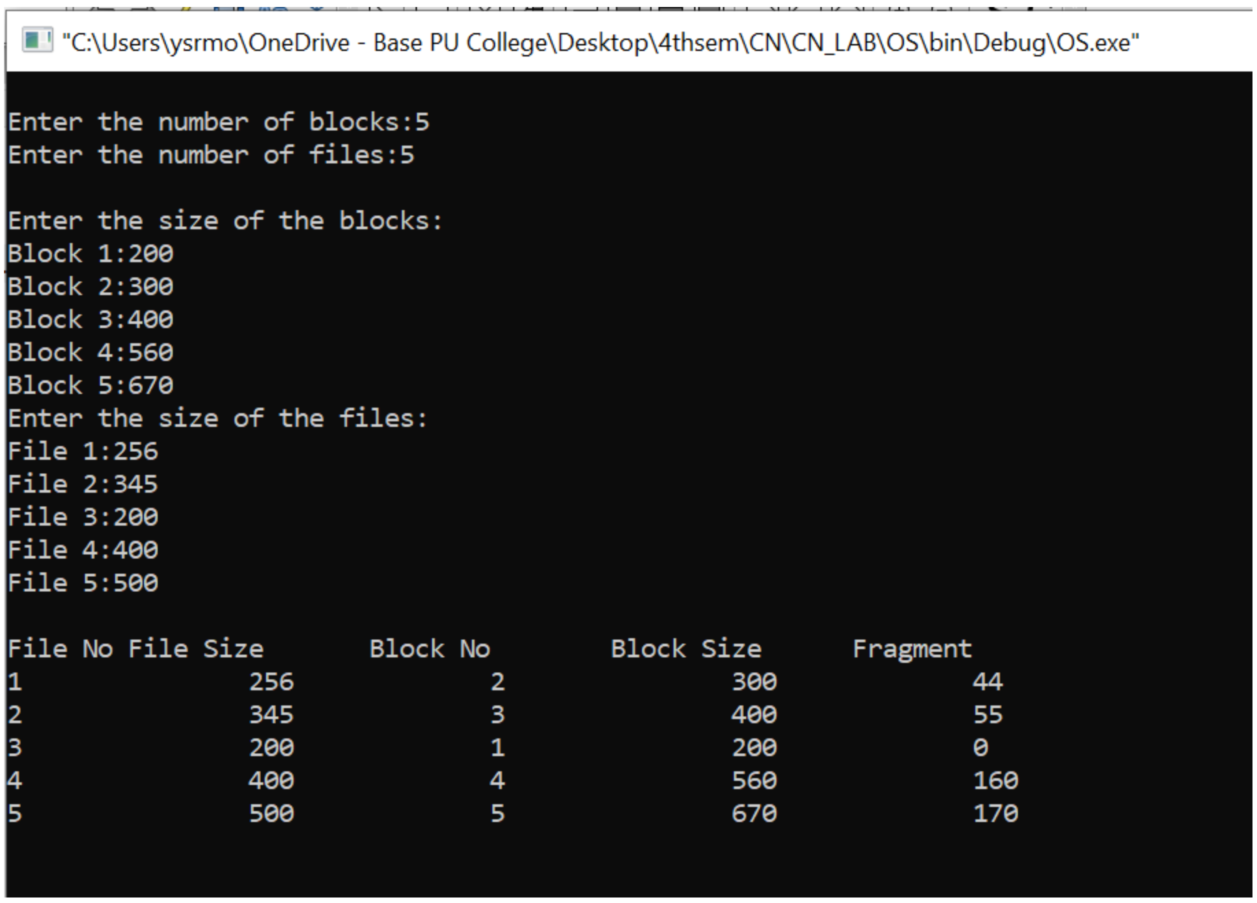
WEEK 9

Write a C program to simulate the best fit contiguous memory allocation technique.

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

#define max 25

void main()
{
   int frag[max], b[max], f[max], i, j, nb, nf, temp, lowest = 10000;
   static int bf[max], ff[max];

   printf("\nEnter the number of blocks:");
   scanf("%d", &nb);
   printf("Enter the number of files:");
   scanf("%d", &nf);

   printf("\nEnter the size of the blocks:\n");
   for (i = 1; i <= nb; i++)
   {
      printf("Block %d:", i);
      scanf("%d", &b[i]);
   }

   printf("Enter the size of the files:\n");
   for (i = 1; i <= nf; i++)
   {
      printf("File %d:", i);
      scanf("%d", &f[i]);
   }
```

```
    for (i = 1; i <= nf; i++)
    {
        lowest = 10000; // Reset lowest to a high value for each new file
        for (j = 1; j <= nb; j++)
        {
            if (bf[j] != 1)
            {
                temp = b[j] - f[i];
                if (temp >= 0 && lowest > temp)
                {
                    ff[i] = j;
                    lowest = temp;
                }
            }
        }
        frag[i] = lowest;
        bf[ff[i]] = 1;
    }

    printf("\nFile No\tFile Size\tBlock No\tBlock Size\tFragment");
    for (i = 1; i <= nf && ff[i] != 0; i++)
    {
        printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d", i, f[i], ff[i], b[ff[i]], frag[i]);
    }

    getch();
}
```

OUTPUT:



"C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\OS\bin\Debug\OS.exe"

```
Enter the number of blocks:5
Enter the number of files:5

Enter the size of the blocks:
Block 1:200
Block 2:300
Block 3:400
Block 4:560
Block 5:670
Enter the size of the files:
File 1:256
File 2:345
File 3:200
File 4:400
File 5:500

File No File Size       Block No        Block Size      Fragment
1               256             2               300             44
2               345             3               400             55
3               200             1               200             0
4               400             4               560             160
5               500             5               670             170
```