FIRST FIT

Aim:

To write a C program for implementation memory allocation methods for fixed partition using first fit.

Algorithm:

1. Define the max as 25.
2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max]. 3: Get the number of blocks,files,size of the blocks using for loop.
4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]
5: Check highest

Program Code:

```
# include <stdio.h>
# define Max 25
int main ()
{ int frag[MAX], b[MAX], f[MAX], bf[MAX], ff[MAX]
  int i, j, nb, nf, temp;
  for(i=0; i<nb; i++)
  {
      scanf("%d", &b[i]);
      bf[i]=0;
      printf(" Enter the no. of files process : ");
      scanf("%d", &nf).
      printf(" Enter the siz of each files :\n");
      for(i=0; i<nf; i++);
      {
          printf("File %d : ", i+1);
          scanf("%d", &f[i]);
      }
  }
}
```

```c
for (i=0; i<nf; i++)
{
    for (j=0; j<nb; j++)
    {
        if (bf[j] == 0 && b[j] >= f[i])
        {
            ff[i] = j;
            bf[j] = 1;
            frag[i] = b[j] - f[i];
            break;
        }
    }
    if (j == nb)
    {
        ff[i] = -1;
    }
}

for (i=0; i<nf; i++)
{
    printf("%d \t \t %d \t \t", i+1, f[i]);
    if (ff[i] != -1)
        b[ff[i], frag[i]];
    else
        printf("Not allocated \t - \t \t \n");
}
```

The fragment of the block are

80
15
23
5
20

Process No	Process-size	Block-no	fragment
P1	20	1	80
P2	30	2	15
P3	50	5	20
P4	40	4	5
P5	10	3	23

The fragment of the block are

Sample Output:

```
Enter the number of blocks:4
Enter the number of files:3

Enter the size of the blocks:-
Block 1:5
Block 2:8
Block 3:4
Block 4:10
Enter the size of the files:-
File 1:1
File 2:4
File 3:7

File_no:        File_size :     Block_no:       Block_size:     Fragment
1               1               1               5               4
2               4               2               8               4
3               7               4               10              3_
```

Result:

Using c the first fit memory allocation algorithm is implemented.