```
#include<stdio.h>

int Psize[10],n,Msize[10],m,i,j,flag;
int psize[100],msize[100],max,loc;

void FIRSTFIT()
{  printf("\n\t\tFIRST FIT\n");
  for(i=0;i<n;i++)
  {  psize[i] = Psize[i];  }
  for(i=0;i<m;i++)
  {  msize[i] = Msize[i];  }
  for(i=0;i<n;i++)
  {  flag=0;
    for(j=0;j<m;j++)
    {  if(msize[j]>=psize[i])
      {  printf("%d ALLOCATED IN %d MEMORY BLOCK",psize[i],msize[j]);
        msize[j] = msize[j]-psize[i];
        printf(" => %d SPACE REMAINING \n",msize[j]);
        flag=1;
        break;            }            }
  if(flag==0)
  {  printf("%d CANNOT BE ALLOCATED \n",psize[i]);    } } }

void WORSTFIT()
{  printf("\n\t\tWORST FIT \n");
  for(i=0;i<n;i++)
  {  psize[i] = Psize[i];  }
  for(i=0;i<m;i++)
  {  msize[i] = Msize[i];  }
  for(i=0;i<n;i++)
  {  max = msize[0];
    loc = 0;
  for(j=0;j<m;j++)
  {  if(msize[j]>max)
    {  max = msize[j];
      loc = j;                    }            }
  if(max>=psize[i])
  {  printf("%d ALLOCATED IN %d MEMORY BLOCK",psize[i],msize[loc]);
    msize[loc] = msize[loc]-psize[i];
    printf(" => %d SPACE REMAINING \n",msize[loc]);   }
  else
  {  printf("%d CANNOT BE ALLOCATED \n",psize[i]);  } } }

void BESTFIT()
{  printf("\n\t\tBEST FIT \n");
  for(i=0;i<n;i++)
  {  psize[i] = Psize[i];  }
  for(i=0;i<m;i++)
  {  msize[i] = Msize[i];  }
  for(i=0;i<n;i++)
  {  loc = -1;
  for(j=0;j<m;j++)
  {  if(msize[j]>=psize[i])
    {  if(loc == -1)
      {  loc = j;  }
    else if(msize[loc]>msize[j])
    {  loc = j;  }    }    }
```

```
  if(loc != -1)
  {  printf("%d ALLOCATED IN %d MEMORY BLOCK",psize[i],msize[loc]);
     msize[loc] = msize[loc]-psize[i];
     printf(" => %d SPACE REMAINING \n",msize[loc]);      }
  else
  {  printf("%d CANNOT BE ALLOCATED \n",psize[i]);  }  }  }

void main()
{  printf("ENTER THE NUMBER OF PROCESS : ");
   scanf("%d",&n);
   printf("ENTER THE ARRAY OF PROCESS : ");
   for(i=0;i<n;i++)
   {  scanf("%d",&Psize[i]);    }
   printf("ENTER THE NUMBER OF MEMORY BLOCK : ");
   scanf("%d",&m);
   printf("ENTER THE ARRAY OF MEMORY BLOCK : ");
   for(i=0;i<m;i++)
   {  scanf("%d",&Msize[i]);   }
   FIRSTFIT();
   BESTFIT();
   WORSTFIT();   }
```