1. **Write a program to show first fit allocation in memory management**

|  |
| --- |
| #include <stdio.h>  #define NUM\_BLOCKS 5  #define NUM\_REQUESTS 7  int blocks[NUM\_BLOCKS];  int requests[NUM\_REQUESTS];  void print\_state() {  int i;  printf("Blocks: ");  for (i = 0; i < NUM\_BLOCKS; i++) {  printf("%d \t", blocks[i]);  }  printf("\nRequests: ");  for (i = 0; i < NUM\_REQUESTS; i++) {  printf("%d \t ", requests[i]);  }  printf("\n");  **}**  **void first\_fit() {**  **int i, j;**  **for (i = 0; i < NUM\_REQUESTS; i++) {**  **for (j = 0; j < NUM\_BLOCKS; j++) {**  **if (requests[i] <= blocks[j]) {**  **printf("Allocated %d to block %d\n", requests[i], j);**  **blocks[j] -= requests[i];**  **break;**  **}**  **}**  **if (j == NUM\_BLOCKS) {**  **printf("Unable to allocate %d\n", requests[i]);**  **}**  **}**  **}**  **int main() {**  **int i;**  **// Initialize the blocks and requests arrays**  **blocks[0] = 100;**  **blocks[1] = 500;**  **blocks[2] = 200;**  **blocks[3] = 300;**  **blocks[4] = 600;**  **requests[0] = 150;**  **requests[1] = 200;**  **requests[2] = 50;**  **requests[3] = 400;**  **requests[4] = 100;**  **requests[5] = 250;**  **requests[6] = 300;**  **// Print the initial state of the system**  **printf("Initial state:\n");**  **print\_state();**  **// Allocate memory using the first fit algorithm**  **first\_fit();**  **// Print the final state of the system**  **printf("Final state:\n");**  **print\_state();**  **return 0;**  **}** |

1. **Write a program to show best fit allocation in memory management:**

|  |
| --- |
| #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++)  {  for(j=1;j<=nb;j++)  {  if(bf[j]!=1)  {  temp=b[j]-f[i];  if(temp>=0)  if(lowest>temp)  {  ff[i]=j;  lowest=temp;  }  }}  frag[i]=lowest;  bf[ff[i]]=1;  lowest=10000;  }  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]);  } |

**2: write a program for implementing sequential file allocation method**

|  |
| --- |
| #include<stdio.h>  #include<stdlib.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();  } |