Slip 16

q.1

#include<stdio.h>

#include<stdlib.h>

//#include<conio.h>

void main()

{

int f[50],i,st,len,j,c,k,count=0;

for(i=0;i<50;i++)

f[i]=0;

printf("File allocation are:\n");

x:count=0;

printf("Enter starting block and length of file:");

scanf("%d%d",&st,&len);

for(k=st;k<(st+len);k++)

if(f[k]==0)

count++;

if(len==count)

{

for(j=st;j<(st+len);j++)

if(f[j]==0)

{

f[j]=1;

printf("%d\t%d\n",j,f[j]);

}

if(j!=(st+len-1))

printf("The file is allocation to disk\n");

}

else

printf("The file is not allocated\n");

printf("Do you want to enter more file(Yes-1/No-0)");

scanf("%d",&c);

if(c==1)

goto x;

else

exit(0);

getch();

}

q.2

#include <stdio.h>

#include <stdlib.h>

#include <mpi.h>

#define ARRAY\_SIZE 1000

int main(int argc, char \*\*argv) {

int rank, size;

int array[ARRAY\_SIZE];

int local\_min,global\_min;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

srand(rank);

for (int i = 0; i < ARRAY\_SIZE / size; i++) {

array[i] = rand();

}

local\_min = array[0];

for (int i = 1; i < ARRAY\_SIZE / size; i++) {

if (array[i] < local\_min) {

local\_min = array[i];

}

}

MPI\_Reduce(&local\_min, &global\_min, 1, MPI\_INT, MPI\_MIN, 0, MPI\_COMM\_WORLD);

if (rank == 0) {

printf("Minimum number: %d\n", global\_min);

}

MPI\_Finalize();

return 0;

}