**WEEK 8**

**Zayd Ahmed**

**1BM21CS254**

**02-08-2023**

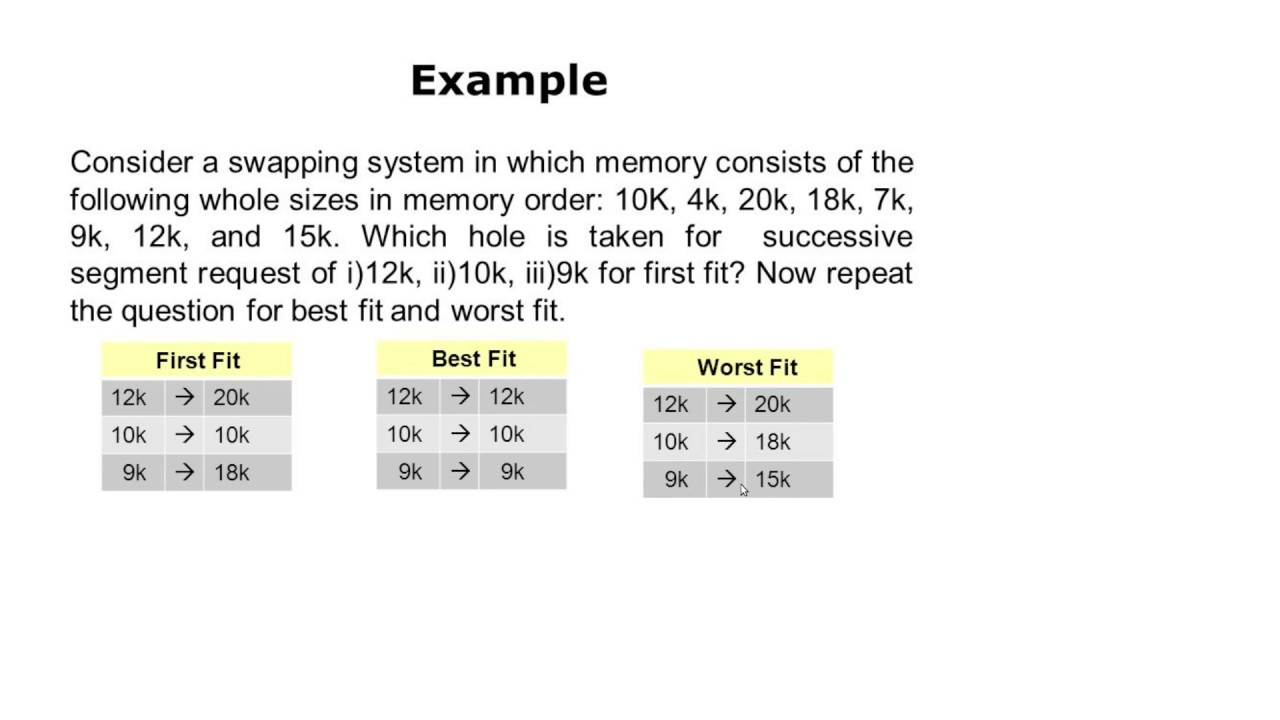
**Q: Write a C program to simulate the following contiguous memory**

**allocation techniques**

**a) Worst-fit**

**b) Best-fit**

**c) First-fit**



#include <stdio.h>

int holes[10];

int holes\_free[10];

int request[10];

int completed[10];

int h,n,temp;

void best\_fit(int n, int h){

for(int i=0;i<n;i++){

completed[i]=0;

}

for(int i=0;i<h;i++){

holes\_free[i]=1;

}

printf("Best Fit:\n");

for(int i=0;i<h-1;i++){

for(int j=0;j<h-i-1;j++){

if(holes[j+1]<holes[j]){

temp=holes[j+1];

holes[j+1]=holes[j];

holes[j]=temp;

}

}

}

for(int i=0;i<n;i++){

for(int j=0;j<h;j++){

if(request[i]<=holes[j] && holes\_free[j]==1 && completed[i]==0){

completed[i]=1;

holes\_free[j]=0;

printf("%dk in %dk\n",request[i],holes[j]);

}

}

}

}

void worst\_fit(int n, int h){

for(int i=0;i<n;i++){

completed[i]=0;

}

for(int i=0;i<h;i++){

holes\_free[i]=1;

}

printf("Worst Fit:\n");

for(int i=0;i<h-1;i++){

for(int j=0;j<h-i-1;j++){

if(holes[j+1]>holes[j]){

temp=holes[j+1];

holes[j+1]=holes[j];

holes[j]=temp;

}

}

}

for(int i=0;i<n;i++){

for(int j=0;j<h;j++){

if(request[i]<=holes[j] && holes\_free[j]==1 && completed[i]==0){

completed[i]=1;

holes\_free[j]=0;

printf("%dk in %dk\n",request[i],holes[j]);

}

}

}

}

void first\_fit(int n, int h){

for(int i=0;i<n;i++){

completed[i]=0;

}

for(int i=0;i<h;i++){

holes\_free[i]=1;

}

printf("First Fit:\n");

for(int i=0;i<n;i++){

for(int j=0;j<h;j++){

if(request[i]<=holes[j] && holes\_free[j]==1 && completed[i]==0){

completed[i]=1;

holes\_free[j]=0;

printf("%dk in %dk\n",request[i],holes[j]);

}

}

}

}

int main(){

printf("enter the number of holes:\t");

scanf("%d",&h);

printf("Enter the holes sizes:\n");

for(int i=0;i<h;i++){

scanf("%d",&holes[i]);

}

printf("enter the number of requests:\t");

scanf("%d",&n);

printf("Enter the request segments:\n");

for(int i=0;i<n;i++){

scanf("%d",&request[i]);

}

best\_fit(n,h);

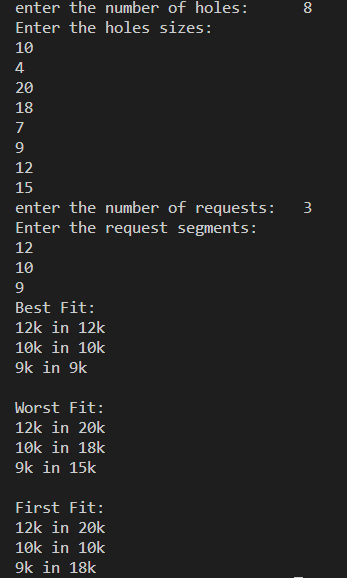
worst\_fit(n,h);

first\_fit(n,h);

return 0;

}

**OUTPUT:**

****