PROJECT CODE :

//C Program for Implementing First, Best and Worst Fir Contiguous memory allocation techniques by keeping free/busy list of jobs organized by memory location.

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct node

{

int start;

int end;

int size;

struct node \*next;

};

struct node \*head = NULL;

struct node \*tail = NULL;

void insert(int start, int end, int size)

{

struct node \*temp = (struct node\*)malloc(sizeof(struct node));

temp->start = start;

temp->end = end;

temp->size = size;

temp->next = NULL;

if(head == NULL)

{

head = temp;

tail = temp;

}

else

{

tail->next = temp;

tail = temp;

}

}

void display()

{

struct node \*temp = head;

while(temp != NULL)

{

printf("%d\t%d\t%d\n", temp->start, temp->end, temp->size);

temp = temp->next;

}

}

void first\_fit(int start, int end, int size)

{

struct node \*temp = head;

while(temp != NULL)

{

if(temp->size >= size)

{

printf("\nJob Allocated at %d\n", temp->start);

temp->start = temp->start + size;

temp->size = temp->size - size;

return;

}

temp = temp->next;

}

printf("\nJob Not Allocated\n");

}

void best\_fit(int start, int end, int size)

{

struct node \*temp = head;

struct node \*best = NULL;

while(temp != NULL)

{

if(temp->size >= size)

{

if(best == NULL)

{

best = temp;

}

else if(best->size > temp->size)

{

best = temp;

}

}

temp = temp->next;

}

if(best != NULL)

{

printf("\nJob Allocated at %d\n", best->start);

best->start = best->start + size;

best->size = best->size - size;

}

else

{

printf("\nJob Not Allocated\n");

}

}

void worst\_fit(int start, int end, int size)

{

struct node \*temp = head;

struct node \*worst = NULL;

while(temp != NULL)

{

if(temp->size >= size)

{

if(worst == NULL)

{

worst = temp;

}

else if(worst->size < temp->size)

{

worst = temp;

}

}

temp = temp->next;

}

if(worst != NULL)

{

printf("\nJob Allocated at %d\n", worst->start);

worst->start = worst->start + size;

worst->size = worst->size - size;

}

else

{

printf("\nJob Not Allocated\n");

}

}

int main()

{

int start, end, size, choice;

char ch;

do

{

printf("\nEnter the start, end and size of the memory block: ");

scanf("%d%d%d", &start, &end, &size);

insert(start, end, size);

printf("\nDo you want to enter more memory blocks? (y/n): ");

scanf(" %c", &ch);

}while(ch == 'y' || ch == 'Y');

printf("\nThe free/busy list of memory blocks is: \n");

display();

do

{

printf("\nEnter the size of the job: ");

scanf("%d", &size);

printf("\n1. First Fit\n2. Best Fit\n3. Worst Fit\nEnter your choice: ");

scanf("%d", &choice);

switch(choice)

{

case 1: first\_fit(start, end, size);

break;

case 2: best\_fit(start, end, size);

break;

case 3: worst\_fit(start, end, size);

break;

default: printf("\nInvalid Choice\n");

}

printf("\nDo you want to enter more jobs? (y/n): ");

scanf(" %c", &ch);

}while(ch == 'y' || ch == 'Y');

printf("\nThe free/busy list of memory blocks is: \n");

display();

return 0;

}