Exp:21

Code:

#include <stdio.h>

#define MAX\_BLOCKS 100

int memory[MAX\_BLOCKS];

int processSize[MAX\_BLOCKS];

void initializeMemory(int size);

int worstFit(int size);

void deallocateMemory(int blockIndex, int size);

void displayMemoryStatus();

int main() {

int memorySize, choice, processSize, blockIndex;

printf("Enter the size of memory: ");

scanf("%d", &memorySize);

initializeMemory(memorySize);

while (1) {

printf("\n1. Allocate Memory\n2. Deallocate Memory\n3. Display Memory Status\n4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter the size of the process: ");

scanf("%d", &processSize);

blockIndex = worstFit(processSize);

if (blockIndex != -1) {

printf("Memory allocated successfully at block %d\n", blockIndex);

} else {

printf("Memory allocation failed. No suitable block found.\n");

}

break;

case 2:

printf("Enter the block index to deallocate: ");

scanf("%d", &blockIndex);

if (blockIndex >= 0 && blockIndex < MAX\_BLOCKS) {

deallocateMemory(blockIndex, memory[blockIndex]);

printf("Memory deallocated successfully from block %d\n", blockIndex);

} else {

printf("Invalid block index.\n");

}

break;

case 3:

displayMemoryStatus();

break;

case 4:

return 0;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

}

return 0;

}

void initializeMemory(int size) {

for (int i = 0; i < MAX\_BLOCKS; i++) {

memory[i] = size;

}

}

int worstFit(int size) {

int maxBlockSize = -1;

int blockIndex = -1;

for (int i = 0; i < MAX\_BLOCKS; i++) {

if (memory[i] >= size && memory[i] > maxBlockSize) {

maxBlockSize = memory[i];

blockIndex = i;

}

}

if (blockIndex != -1) {

memory[blockIndex] -= size;

}

return blockIndex;

}

void deallocateMemory(int blockIndex, int size) {

memory[blockIndex] += size;

}

void displayMemoryStatus() {

printf("Memory Status:\n");

for (int i = 0; i < MAX\_BLOCKS; i++) {

printf("Block %d: %d\n", i, memory[i]);

}

}

Output:

