WEEK-6

Huffman Coding & Job Sequencing Problem

Huffman Coding:

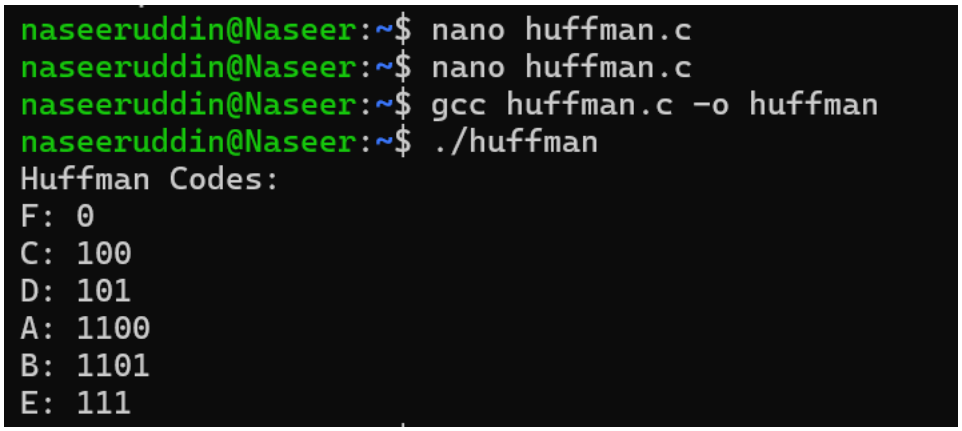
CODE:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100

struct Node{
   char data;
unsigned freq;
   struct Node *left,*right;
};

struct Node* newNode(char data,unsigned freq){
   struct Node* temp=(struct Node*)malloc(sizeof(struct Node));
   temp->left=temp->right=NULL;
   temp->data=data;
   temp->freq=freq;
   return temp;
}

struct MinHeap{
   unsigned size;
   struct Node* arr[MAX];
};

void swap(struct Node** a,struct Node** b){
   struct Node* t=*a;
   *a=*b;
   *b=t;
}

void heapify(struct MinHeap* heap,int i){
   int smallest=i;
   int left=2*i+1;
   int right=2*i+2;
```

```
    if(left<heap->size && heap->arr[left]->freq<heap->arr[smallest]->freq)
        smallest=left;

    if(right<heap->size && heap->arr[right]->freq<heap->arr[smallest]->freq)
        smallest=right;

    if(smallest!=i){
        swap(&heap->arr[i],&heap->arr[smallest]);
        heapify(heap,smallest);
    }
}

struct Node* extractMin(struct MinHeap* heap){
    struct Node* temp=heap->arr[0];
    heap->arr[0]=heap->arr[--heap->size];
    heapify(heap,0);
    return temp;
}

void insertHeap(struct MinHeap* heap,struct Node* node){
    int i=heap->size++;
    while(i && node->freq<heap->arr[(i-1)/2]->freq){
        heap->arr[i]=heap->arr[(i-1)/2];
        i=(i-1)/2;
    }
    heap->arr[i]=node;
}

struct Node* buildTree(char data[],int freq[],int size){
    struct MinHeap heap;
    heap.size=0;

    for(int i=0;i<size;i++)
        heap.arr[heap.size++]=newNode(data[i],freq[i]);

    for(int i=(heap.size-1)/2;i>=0;i--)
        heapify(&heap,i);

    while(heap.size>1){
        struct Node* left=extractMin(&heap);
```

```
        struct Node* right=extractMin(&heap);

        struct Node* top=newNode('$',left->freq+right->freq);
        top->left=left;
        top->right=right;

        insertHeap(&heap,top);
    }
    return extractMin(&heap);
}

void printCodes(struct Node* root,int arr[],int top){
    if(root->left){
        arr[top]=0;
        printCodes(root->left,arr,top+1);
    }
    if(root->right){
        arr[top]=1;
        printCodes(root->right,arr,top+1);
    }
    if(!root->left && !root->right){
        printf("%c: ",root->data);
        for(int i=0;i<top;i++)
            printf("%d",arr[i]);
        printf("\n");
    }
}

int main(){
    char data[]={'A','B','C','D','E','F'};
    int freq[]={5,9,12,13,16,45};
    int size=sizeof(data)/sizeof(data[0]);

    struct Node* root=buildTree(data,freq,size);

    int arr[MAX],top=0;
    printf("Huffman Codes:\n");
    printCodes(root,arr,top);

    return 0;
}
```

OUTPUT:



Job Sequencing Problem

CODE:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
int id;
int deadline;
int profit;
} Job;

int compare(const void *a, const void *b) {
Job *j1 = (Job *)a;
Job *j2 = (Job *)b;
return j2->profit - j1->profit;
}

int main() {
int n;
printf("Enter number of jobs: ");
scanf("%d", &n);

Job jobs[n];

for(int i = 0; i < n; i++) {
printf("Enter deadline and profit for job %d: ", i+1);
```

```
scanf("%d %d", &jobs[i].deadline, &jobs[i].profit);
jobs[i].id = i + 1;
}

qsort(jobs, n, sizeof(Job), compare);

int maxDeadline = 0;
for(int i = 0; i < n; i++)
if(jobs[i].deadline > maxDeadline)
maxDeadline = jobs[i].deadline;

int slot[maxDeadline];
for(int i = 0; i < maxDeadline; i++)
slot[i] = -1;

int totalProfit = 0;

for(int i = 0; i < n; i++) {
for(int j = jobs[i].deadline - 1; j >= 0; j--) {
if(slot[j] == -1) {
slot[j] = jobs[i].id;
totalProfit += jobs[i].profit;
break;
}
}
}


printf("\nScheduled Jobs: ");
for(int i = 0; i < maxDeadline; i++)
if(slot[i] != -1)
printf("J%d ", slot[i]);

printf("\nTotal Profit = %d\n", totalProfit);

return 0;
}
```

OUTPUT:

```
naseeruddin@Naseer:~$ nano sequencing.c
naseeruddin@Naseer:~$ gcc sequencing.c -o sequencing
naseeruddin@Naseer:~$ ./sequencing
Enter number of jobs: 4
Enter deadline and profit for job 1: 4 20
Enter deadline and profit for job 2: 1 10
Enter deadline and profit for job 3: 1 40
Enter deadline and profit for job 4: 1 30

Scheduled Jobs: J3 J1
Total Profit = 60
```