

```
#include<string.h>

#include<stdio.h>

#include<stdlib.h>

typedef struct node
{
    char ch;

    int freq;

    struct node *left;

    struct node *right;
}node;

node * heap[100];

int heapSize=0;

void Insert(node * element)
{
    heapSize++;

    heap[heapSize] = element;

    int now = heapSize;

    while(heap[now/2] -> freq > element -> freq)
    {
        heap[now] = heap[now/2];

        now /= 2;
    }

    heap[now] = element;
```

```

}

node * DeleteMin()
{
    node * minElement,*lastElement;

    int child,now;

    minElement = heap[1];

    lastElement = heap[heapSize--];

    for(now = 1; now*2 <= heapSize ;now = child)
    {
        child = now*2;

        if(child != heapSize && heap[child+1]->freq < heap[child] -> freq )
        {
            child++;
        }

        if(lastElement -> freq > heap[child] -> freq)
        {
            heap[now] = heap[child];
        }
        else
        {
            break;
        }
    }

    heap[now] = lastElement;

    return minElement;
}

```

```

}

void print(node *temp,char *code)
{
    if(temp->left==NULL && temp->right==NULL)
    {
        printf("char %c code %s\n",temp->ch,code);
        return;
    }

    int length = strlen(code);
    char leftcode[10],rightcode[10];
    strcpy(leftcode,code);
    strcpy(rightcode,code);
    leftcode[length] = '0';
    leftcode[length+1] = '\0';
    rightcode[length] = '1';
    rightcode[length+1] = '\0';
    print(temp->left,leftcode);
    print(temp->right,rightcode);
}

int main()
{

    heap[0] = (node *)malloc(sizeof(node));

    heap[0]->freq = 0;

    int n ;

```

```

printf("Enter the no of characters: ");

scanf("%d",&n);

printf("Enter the characters and their frequencies: ");

char ch;

int freq,i;

for(i=0;i<n;i++)
{
    scanf(" %c",&ch);

    scanf("%d",&freq);

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

    temp -> ch = ch;

    temp -> freq = freq;

    temp -> left = temp -> right = NULL;

    Insert(temp);
}

if(n==1)
{
    printf("char %c code 0\n",ch);

    return 0;
}

for(i=0;i<n-1 ;i++)
{
    node * left = DeleteMin();

    node * right = DeleteMin();
}

```

```
node * temp = (node *) malloc(sizeof(node));

temp -> ch = 0;

temp -> left = left;

temp -> right = right;

temp -> freq = left->freq + right -> freq;

Insert(temp);

}

node *tree = DeleteMin();

char code[10];

code[0] = '\0';

print(tree,code);

}
```