

1

```
# include < stdio.h >
# define MAX_NAME 30
# define MAX_DEP 5
```

```
Struct Traveler {
```

```
    int passportID;
```

```
    char name[MAX_NAME];
```

```
    int riskScore;
```

```
    int numDependents;
```

```
    struct Traveler *dependents[MAX_DEPENDENTS];
```

```
}
```

```
Struct Traveler t;
```

```
Void iscleared ( struct Traveler *t ) {
```

```
    if ( t->riskScore >= 50 )
```

```
        return 0;
```

```
    for ( int i=0 ; i < t->numDependents ; i++ ) {
```

```
        if ( t->dependents[i]->riskScore >= 50 )
```

```
            return 0;
```

```
}
```

```
    else
```

```
        return 1;
```

```
Void printHighRiskMembers ( struct Traveler * t ) {
```

```
if ( t -> riskScore >= 70 )
```

```
    printf (" passport ID: %d | Name: %s \n",  
           t -> passportID, t -> name );
```

```
for ( int i = 0 ; i < t -> numDependents ; i++ ) {
```

```
if ( t -> dependents [i] -> riskScore >= 70 )
```

```
    printf (" passport ID: %d | Name: %s \n",  
           t -> dependents [i] -> riskScore, t -> dependents  
           [i] -> name );
```

```
else return;
```

```
}
```

```
}
```

```
int calculateTotalRisk ( struct Traveler * t ) {
```

```
if ( t == NULL ) return 0;
```

```
int sum = t -> riskScore;
```

```
for ( int i = 0 ; i < t -> numDependents ; i++ ) {
```

```
    sum += calculateTotalRisk ( t -> dependents [i] );
```

```
}
```

```
return sum;
```

```
}
```

Void EnterTraveler (struct Traveler *t) {

printf ("Enter Traveler Data\n");

printf ("ID :- ");

scanf ("%d", &(t->passportID));

printf ("Name :- ");

scanf ("%s", &(t->name));

printf ("RiskScore :- ");

scanf ("%d", &(t->riskScore));

printf ("NumDepends, &(t->numDepends));

for (int i=0; i < t->numDepends; i++) {

t->dependents[i] = malloc(sizeof(struct Traveler));

printf ("ID :- ");

scanf ("%d", &(t->dependents[i]->passportID));

printf ("Name :- ");

scanf ("%s", &(t->dependents[i]->name));

printf ("RiskScore :- ");

scanf ("%d", &(t->dependents[i]->riskScore));

}

}

```
int Main() {  
    printf ("Welcome");  
    struct Traveler *t = malloc (sizeof (struct Traveler));  
    int n;  
    printf ("Enter 1 for Input \n");  
    scanf ("%d", &n);  
    switch (n) {  
        case 1:  
            EnterTraveler (t);  
            break;  
        case default:  
            break;  
    }  
    if (isCleared (t)) printf ("Cleared");  
    else printf ("DENIED");  
    printHighRiskMembers (t);  
    printf ("Total Risk :- %.d", calculateTotalRisk (t));  
    return 0;  
}
```

2

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

```
int totalCost = 0;
```

```
void mergeFiles ( int arr[], int n ) {
```

```
if ( n == 1 )
```

```
return;
```

```
int minIndex = 0;
```

```
int minSum = arr[0] + arr[1];
```

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

```
if ( arr[i] + arr[i + 1] < minSum ) {
```

```
minSum = arr[i] + arr[i + 1];
```

```
minIndex = i;
```

```
}
```

```
}
```

```
printf ( " Merge %.d and %.d is %.d (cost: %.d) \n " ,  
arr[minIndex], arr[minIndex + 1], minSum, minSum );
```

```
totalCost += minSum;
```

```
arr[minIndex] = minSum;
```

```
for ( int i = 0 , j = i < n - 1 ; i++ ) {
```

```
minIndex + 1
```

```
arr[i] = arr[i + 1];
```

```
}
```

```
mergeFiles ( arr, n - 1 );
```

```
}
```

```
int main () {
```

```
    int n;
```

```
    printf ("Enter n:-");
```

```
    scanf ("%d", &n);
```

```
    arr[n];
```

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

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

```
}
```

```
    mergefiles (arr, n);
```

```
    printf ("Final file size: %d \n", arr[0]);
```

```
    printf ("Total Merge cost: %d \n", totalCost);
```

```
    return 0;
```

```
}
```