

**Aim:**

Write a C program to perform optimal merging on a given input array of elements, and print the output as shown in the examples.

**Source Code:****OptimalMerge.c**

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

// Function to Sort the files in ascending order, perform optimal file merging and re
return the minimum cost
int optimalMerge(int files[], int n) {

    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if (files[j] < files[j+1]){
                int temp = files[j];
                files[j]=files[j+1];
                files[j+1] = temp;
            }
        }
    }
    int cost=0;
    while(n>1){
        int merge = files[0]+files[1];
        cost += merge;
        files[0] = merge;
        for(int i=1;i<n-1;i++){
            files[i]=files[i+1];
        }
        n--;
        for (int i=0;i<n-1;i++){
            for (int j=0; j<n-i-1;j++){
                if (files[j] > files[j+1]){
                    int temp = files[j];
                    files[j]=files[j+1];
                    files[j+1]=temp;
                }
            }
        }
    }
    return cost;
}

int main() {
    int n;
    printf("Number of files: ");
    scanf("%d", &n);
    int *files = (int *)malloc(n * sizeof(int));
    printf("Enter the sizes of %d files: ", n);
    for (int i = 0; i < n; i++) {
```

```

        scanf("%d", &files[i]);
    }
    int minCost = optimalMerge(files, n);
    printf("Minimum cost of merging is: %d\n", minCost);
    free(files);
    return 0;
}

```

### Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Number of files: 5
Enter the sizes of 5 files: 20 10 5 30 30
Minimum cost of merging is: 205

Test Case - 2
User Output
Number of files: 6
Enter the sizes of 6 files: 8 11 16 18 9 20
Minimum cost of merging is: 208