|  |  |
| --- | --- |
| **Ex No: 6** | Find Sum of Weights and Sorting |

**AIM**

Given a set of numbers like <10, 36, 54, 89, 12, 27>, find sum of weights based on the following conditions.

5 if it is a perfect cube.

4 if it is a multiple of 4 and divisible by 6.

3 if it is a prime number.

Sort the numbers based on the weight in the increasing order as shown below

<10,its weight>,<36,its weight><89,its weight>

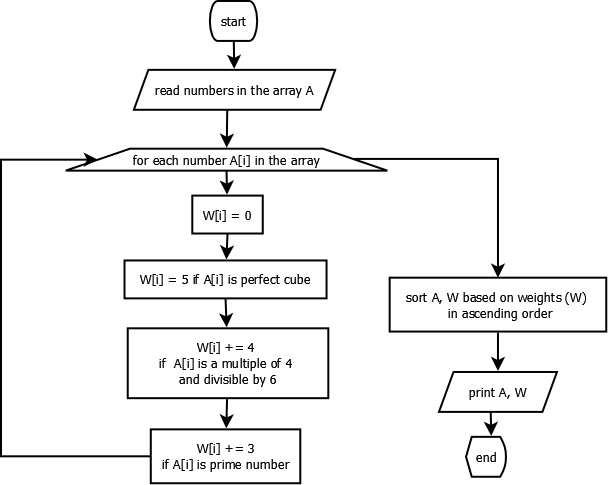
**PRE-LAB QUESTIONS**

1. How do you pass an array to C function?
2. Differentiate puts() and putchar()
3. What is sorting
4. List few sorting methods
5. What is merge sort

**ALGORITHM**

**Step 1:** Start  
**Step 2:**  read numbers in the array A  
**Step 3:** For each number A[i] in the array:  
 Step 3.1: W[i] = 0  
 Step 3.2: Add **5** to W[i] if it is a perfect cube.  
 Step 3.3: Add **4** to W[i] if .  
 Step 3.4: Add **3** to W[i] if it is a prime number.  
**Step 4:**  Sort the numbers (A, W) based on weights (W) in the increasing order  
**Step 5:** End

**FLOWCHART**



**PROGRAM**

#include "stdio.h"

#define SIZE 5

int main**(){**

int A**[**SIZE**];**

int W**[**SIZE**];**

int i**,** k**,** prime**,** t**;**

/\* read numbers \*/

**for(**i**=**0**;** i **<** SIZE**;** i**++){**

scanf**(**"%d"**,** **&**A**[**i**]);**

**}**

/\* find weights \*/

**for(**i**=**0**;** i **<** SIZE**;** i**++){**

W**[**i**]** **=** 0**;**

// check if A[i] is perfect cube

**for(**k **=** 1 **;** k **<** A**[**i**]** **/** 2**;** k**++** **)**

**{**

**if** **(**k**\***k**\***k **==** A**[**i**])**

**{**

W**[**i**]** **=** 5**;**

**break;**

**}**

**}**

// multiple of 4 and divisible by 6

**if** **(**A**[**i**]** **%** 4 **==** 0 **&&** A**[**i**]** **%** 6 **==** 0 **)**

W**[**i**]** **+=** 4**;**

// check if A[i] is prime number

prime **=** 1**;**

**for(**k **=** 2**;** k **<** A**[**i**];** k**++)**

**{**

**if** **(**A**[**i**]** **%** k **==** 0**)**

**{**

prime **=** 0**;**

**break;**

**}**

**}**

**if(**prime **==** 1**)**

W**[**i**]** **+=** 3**;**

printf**(**"<%d, %d>, "**,** A**[**i**],** W**[**i**]);**

**}**

/\* sorting based on W \*/

**for(**i**=**0**;** i **<** SIZE**;** i**++)**

**for(**k**=**i**+**1**;** k **<** SIZE**;** k**++)**

**if** **(**W**[**i**]** **>** W**[**k**])**

**{**

t **=** A**[**i**];** A**[**i**]** **=** A**[**k**];** A**[**k**]** **=** t**;**

t **=** W**[**i**];** W**[**i**]** **=** W**[**k**];** W**[**k**]** **=** t**;**

**}**

/\* print after sorting \*/

printf**(**"\n After sorting based on weights \n"**);**

**for(**i**=**0**;** i **<** SIZE**;** i**++)**

printf**(**"<%d, %d>, "**,** A**[**i**],** W**[**i**]);**

printf**(**"\n"**);**

**}**

**INPUT**

3 9 27 24 21

**OUTPUT**

<3, 3>, <9, 0>, <27, 5>, <24, 4>, <21, 0>,

After sorting based on weights

<9, 0>, <21, 0>, <3, 3>, <24, 4>, <27, 5>,

**POST-LAB QUESTIONS**

1. Find the maximum element in the array.
2. Search key K in the given array of numbers A using binary search

**RESULT**

Thus the C program to find the program to find the sum of weights using given conditions was successfully written and executed.