#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\*\* split\_string(char\*);

// Complete the miniMaxSum function below.

void miniMaxSum(int arr\_count, int\* arr) {

long long int temp;

long long int i,j;

//sort the array

for(i=0;i<arr\_count;i++)

{

for(j=0;j<arr\_count;j++)

{

if(arr[i]>arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("%lld",(arr[4]+arr[1]+arr[2]+arr[3]));

printf(" %lld",(arr[0]+arr[1]+arr[2]+arr[3]));

}

int main()

{

char\*\* arr\_temp = split\_string(readline());

int\* arr = malloc(5 \* sizeof(int));

for (int i = 0; i < 5; i++) {

char\* arr\_item\_endptr;

char\* arr\_item\_str = \*(arr\_temp + i);

int arr\_item = strtol(arr\_item\_str, &arr\_item\_endptr, 10);

if (arr\_item\_endptr == arr\_item\_str || \*arr\_item\_endptr != '\0') { exit(EXIT\_FAILURE); }

\*(arr + i) = arr\_item;

}

int arr\_count = 5;

miniMaxSum(arr\_count, arr);

return 0;

}

char\* readline() {

size\_t alloc\_length = 1024;

size\_t data\_length = 0;

char\* data = malloc(alloc\_length);

while (true) {

char\* cursor = data + data\_length;

char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

if (!line) { break; }

data\_length += strlen(cursor);

if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') { break; }

size\_t new\_length = alloc\_length << 1;

data = realloc(data, new\_length);

if (!data) { break; }

alloc\_length = new\_length;

}

if (data[data\_length - 1] == '\n') {

data[data\_length - 1] = '\0';

}

data = realloc(data, data\_length);

return data;

}

char\*\* split\_string(char\* str) {

char\*\* splits = NULL;

char\* token = strtok(str, " ");

int spaces = 0;

while (token) {

splits = realloc(splits, sizeof(char\*) \* ++spaces);

if (!splits) {

return splits;

}

splits[spaces - 1] = token;

token = strtok(NULL, " ");

}

return splits;

}