

KẾT QUẢ BÀI TẬP VỀ NHÀ

I. Bài làm của các nhóm

1. Nhóm 1

```
from math import prod

def max_prod(lst: list) -> int:
    pos = [i for i in lst if i > 0]
    neg = [i for i in lst if i < 0]

    if len(neg) % 2:
        neg.remove(max(neg))

    if len(pos) > 0 or len(neg) > 0:
        return prod(pos) * prod(neg)

    return 0

if __name__ == '__main__':
    n = int(input())
    lst = list(map(int, input().split()))
    print(max_prod(lst))
```

2. Nhóm 2

```
def maxProductSubset(a, n):
    if n == 1:
        return a[0]

    max_neg = -999999999999
    count_neg = 0
```

```

count_zero = 0
prod = 1
for i in range(n):
    if a[i] == 0:
        count_zero += 1
        continue

    if a[i] < 0:
        count_neg += 1
        max_neg = max(max_neg, a[i])

    prod = prod * a[i]
if count_zero == n:
    return 0

if count_neg & 1:
    if (count_neg == 1 and count_zero > 0 and
        count_zero + count_neg == n):
        return 0
    prod = int(prod / max_neg)

return prod

if __name__ == '__main__':
    n=int(input())
    a = [int(x) for x in input().split()]
    print(maxProductSubset(a, n))

```

3. Nhóm 3

```

n = int(input())
arr = list(map(int, input().split()))
sorted(arr)

```

```

def Max_Product(n, arr):
    negative = [i for i in arr if i < 0]
    positive = [i for i in arr if i > 0]

    if len(negative) % 2 != 0:
        negative.remove(max(negative))

    if len(positive) == 0 and len(negative) == 0:
        return 0
    else:
        product = 1
        for j in negative:
            product *= j
        for k in positive:
            product *= k
        return product

print(Max_Product(n, arr))

```

4. Nhóm 4

```

def solve(a,length):
    result = 1
    allZero = True
    for i in range(len(a)):
        if a[i] > 0:
            allZero = False
            result *= a[i]
        elif a[i] < 0:
            allZero = False
            if (i + 1 == length and result > 0) or (i + 1 < length and
a[i + 1] >= 0 and result > 0):

```

```

        continue
    result *= a[i]
else:
    allZero = allZero and (a[i] == 0)
return result if not allZero else 0

length = int(input())
a = list(map(int,input().split()))

if length == 1:
    print(a[0])
else:
    a = sorted(a)
    print(solve(a,length))

```

5. Nhóm 5

```

def maxMultiplication(a):
    s = 1
    arr = [x for x in a if x < 0]
    arr0 = [x for x in a if x == 0]
    if len(arr) != 0:
        maxNegative = max(arr)
    if len(arr0) == len(a):
        return 0
    if len(arr0) + len(arr) == len(a) and len(arr) == 1:
        return arr[0]
    for item in a:
        if item == 0:
            continue
        else:

```

```

        s *= item
    if s < 0:
        s //= maxNegative
    return s

if __name__ == '__main__':
    n = int(input())
    a = [int(x) for x in input().split()]
    print(maxMultiplication(a))

```

6. Nhóm 6

```

def maxSubArray(A): # max sub array product
    totalMax = A[0] # total max
    currentMin = A[0] # current min
    currentMax = A[0]
    current = A[0]
    for i in range (1,len(A)):
        current = A[i];
        maxTemp = max(currentMin*current, currentMax*current, current)
        minTemp = min(currentMin*current, currentMax*current, current)
        currentMin = minTemp
        currentMax = maxTemp
        if currentMax > totalMax:
            totalMax=currentMax
    return totalMax

# if main

n = int(input())
data = list(map(int,input().split()))
print (maxSubArray(data))

```

7. Nhóm 7

```
def maxMultiplication(a):
    s = 1
    arr = [x for x in a if x < 0]
    arr0 = [x for x in a if x == 0]
    if len(arr) != 0:
        maxNegative = max(arr)
    if len(arr0) == len(a):
        return 0
    if len(arr0) + len(arr) == len(a) and len(arr) == 1:
        return arr[0]
    for item in a:
        if item == 0:
            continue
        else:
            s *= item
    if s < 0:
        s //= maxNegative
    return s

if __name__ == '__main__':
    n = int(input())
    a = [int(x) for x in input().split()]
    print(maxMultiplication(a))
```

8. Nhóm 9

```
import sys

class MaximumProdOfSubSetProblem:
    def __init__(self, quantity, array):
        self.sizeOfArray = quantity
```

```
self.array = array

def solution(self):
    if self.sizeOfArray == 1:
        return self.array[0]

    maxNev = -sys.maxsize
    countZero = countNev = 0
    maxProd = 1

    for element in self.array:
        if element == 0:
            countZero += 1
            continue

        if element < 0:
            countNev += 1
            maxNev = max(maxNev, element)

        maxProd *= element

    if countZero == self.sizeOfArray:
        return 0

    if countZero == self.sizeOfArray - 1 and countNev == 1:
        return 0

    if maxProd < 0:
        maxProd //= maxNev

    return maxProd
```

```

if __name__ == "__main__":
    quantity = int(input())
    array = list([int(element) for element in input().split()])

    print(MaximumProdOfSubSetProblem(quantity, array).solution())

```

9. Nhóm 10

```

def Tich(data, n):
    neg = [i for i in data if i < 0]
    pos = [i for i in data if i > 0]
    if n == 1:
        return data[0]
    elif n == 0:
        return 0
    if len(neg) % 2 != 0:
        neg.remove(max(neg))
    if len(pos) == 0 and len(neg) == 0:
        return 0
    tich = 1
    for x in neg:
        tich *= x
    for x in pos:
        tich *= x
    return tich

n = int(input())
data = list(map(int, input().split()))
sorted(data)
print(Tich(data, n))

```


10.Nhóm 11

KHÔNG NỘP

11.Nhóm 12

```
def Max_pro(arr, l):
    if l == 1:
        return arr[0]
    elif l == 0:
        return 0
    neg = [i for i in arr if i < 0]
    pos = [i for i in arr if i > 0]
    if len(neg) % 2 != 0:
        neg.remove(max(neg))
    if len(pos) == 0 and len(neg) == 0:
        return 0
    pro = 1
    for j in neg:
        pro *= j
    for j in pos:
        pro *= j
    return pro

l = int(input())
arr = list(map(int, input().split()))
sorted(arr)
print(Max_pro(arr, l))
```

12.Nhóm 13

```
n = int(input())
a = [int(x) for x in input().split()]
```

```

zero, neg, pos = list(), list(), list()
res = 1

if len(a) == 1:
    print(a[0])
else:
    for i in a:
        if i == 0:
            zero.append(i)
        elif i < 0:
            neg.append(i)
            res *= i
        else:
            pos.append(i)
            res *= i
    if len(zero) == (n - 1) and len(neg) == 1:
        print(0)
    elif len(zero) == (n - 2) and len(neg) == 1 and len(pos) == 1:
        print(0)
    elif len(zero) == n:
        print(0)
    else:
        if res < 0:
            res //= max(neg)
            print(res)
        else:
            print(res)

```

13.Nhóm 14

```

n = int(input())
lst = [int(x) for x in input().split()]
lst1 = [x for x in lst if x < 0]

```

```
lst2 = [x for x in lst if x > 0]
lst3 = [x for x in lst if x == 0]
```

```
def Tich(lst):
    tich = 1
    for i in range(len(lst)):
        tich *= lst[i]
    return tich

t1, t2 = Tich(lst1), Tich(lst2)
if t1 < 0:
    t1 //= max(lst1)
if len(lst) == 1:
    print(lst[0])
elif lst2:
    print(t1 * t2)
elif t1 > 0 and len(lst1) > 1:
    print(t1)
else: print(0)
```

14.Nhóm 15

```
def product (arr):
    result = 1
    for i in arr:
        result *= i
    return result

def max_(arr):
    arr_neg = []
    arr_pos = []
    for x in arr:
        if x < 0:
```

```

        arr_neg.append(x)
    elif x > 0:
        arr_pos.append(x)

    pro_neg = product(arr_neg)
    pro_pos = product(arr_pos)

    if pro_neg < 0:
        pro_neg //= max(arr_neg)
    if len(arr) == 1:
        return arr[0]
    elif arr_pos != None:
        return pro_neg*pro_pos
    elif pro_neg > 0 and len(arr_pos) > 1:
        return pro_neg
    else: return 0

if __name__ == '__main__':
    n = int(input())
    arr = [int(x) for x in input().split()]
    print(max_(arr))

```

15. Tổng kết:

- Số nhóm giải đúng: 12
- Số nhóm chưa giải đúng: 1 (90/100)
- Số nhóm không nộp bài: 1

II. Nhận xét

Các hướng làm bài chính:

- Chia các phần tử vào các mảng (mảng dương, mảng âm, mảng 0), sau đó thao tác trên các mảng.
- Xét từng phần tử trong mảng, tính tích tại mỗi lần xét: Nhóm 9, Nhóm 4, Nhóm 2
- Nhóm 4 đặc biệt, có biến allZero, và cần sắp xếp mảng
- Nhóm 6 có cách giải khác