In [10]:

```python
# coding for the closest value of the zero if there are so many values we have to
n=int(input())
s=input()
s=s.split()
# "1 2 3 4 5"
li=[]
for i in s:
    li.append(abs(int(i)))
min(li)
```

```
5
-1 2 4 7 3
```

Out[10]:

```
1
```

In [17]:

```
n=int(input())
s=input()
s=s.split()
# "1 2 3 4 5"
li=[]
for i in s:
    li.append(abs(int(i)))

#for i in range (0,max(li)):

def closestZero(li):
    if 0 in li:
        return 0
    else:
        li.sort()
        pc = []
        nc=  []
        for i in li:
            if i>0:
                pc.append(i)
        for i in li:
            if i<0:
                nc.append(i)
        if len(nc)==0:
            return min(pc)
        elif len(pc)==0:
            return max(nc)
        else:
            nz=max(nc)
            pz=min(pc)
            if abs(nz) > pc or abs(nc)==pc:
                return pc
            else:
                return nz
closestZero(li)
```

```
6
-1 1 2 -2 300 -300
```

Out[17]:

```
1
```

In [1]:

```
# Function to find the no of hours in a given time period

def isLeapYear(year): # To check if a given year is a Leap Year
    if year % 400 == 0 or (year % 100 != 0 and year % 4 == 0):
        return True
    return False




def numberOfDays(startyear, endyear):
    sum = 0
    for year in range(startyear, endyear+1):
        if isLeapYear(year):
            sum = sum + 366
        else:
            sum = sum + 365
    return sum
#number of days in middle years of 2016 2019
numberOfDays(2017, 2018)

def numberOfDaysMonth(month, year):
    if month == 2:
        if isLeapYear(year):
            return 29
        return 28
    elif (month <= 7 and month % 2!= 0) or (month >= 8 and month % 2 == 0):
        return 31
    else:
        return 30

def daysInStartYear(startmonth, startyear):
    days = 0
    for month in range(startmonth, 13):
        days += numberOfDaysMonth(month, startyear)
    return days

def daysInEndYear(endmonth, endyear):
    days = 0
    for month in range(1, endmonth+1):
        days += numberOfDaysMonth(month, endyear)
    return days

def numberOfHours(startmonth, startyear, endmonth, endyear):
    days = 0
    if startyear != endyear:
        days += daysInStartYear(startmonth, startyear)
        days += daysInEndYear(endmonth, endyear)
        if endyear - startyear == 2: # 2019 - 2017
            days += numberOfDays(startyear+1, startyear+1)
        elif endyear - startyear > 2:
            days += numberOfDays(startyear+1, endyear-1)
    else:
        for month in range(startmonth, endmonth+1):
            days += numberOfDaysMonth(month, startyear)
    return 24 * days
```

```
numberOfHours(11, 1975,3, 2018)
```

Out[1]:

```
371808
```

In [ ]:

```
P
```