과목 명: 시스템프로그래밍

담당 교수 명: 소 정 민

<<Assignment 4>>

**서강대학교 컴퓨터공학과**

**[20151547]**

**[박재상]**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 4

3. 모듈 정의 5

3.1 모듈 이름 : load1(char\*) 5

3.1.1 기능 load a single program 5

3.1.2 사용 변수 char\* input : program file name 5

3.2 모듈 이름: load2(char\*,char\*) 5

3.2.1 기능 load two programs 5

3.2.2 사용 변수 char\* input1, char\*input2 5

3.3 모듈이름:load3(char\*,char\*,char\*) 5

3.3.1 기능 load three programs 5

3.3.2 사용변수 char\* input1, char\*input2,char\* input3 5

3.4 모듈이름:write(char\*,char\*) 5

3.4.1 기능write objectcode on memory 5

3.4.2 사용변수 char\* from, char\* length 5

5. 코드 7~28

# 프로그램 개요

; 해당 Assignment의 프로그램에 대한 기능 및 특이점에 대한 전반적인 개요에 대해서 설명한다.

* 1. 파이썬3를 이용하여 코드를 작성하였다.
  2. 모든 작업은 colab에서 진행하였다.
  3. 사용자가 직접 input값을 결정하며, output은 모니터를 통해 출력된다.
  4. 총 4개의 문제를 풀기위한 코드를 작성하여 이에 대한 파일을 만들었다.

# 프로그램 설명

## Max counters

사용자가 최대값 N을 입력한 뒤, 원하는 숫자열 A[0,M]을 입력한다.

그리고 counter를 적용할 숫자열 B[0,M]를 [0,0,…..,0]으로 초기화한다.

숫자열 A[0,M]를 첫 번째 index부터 마지막 index까지 한 번씩 읽어나간다.

A[i] (0<= i<M) 값이 N보다 작거나 같을 경우 B[i]값이 1 증가한다.

A[i]값이 N보다 클 경우에는 숫자열 B[0,M]의 값들 중 최대값 max를 찾아서 B[0, M]를 [max,max,…,max]로 초기화한다.

이 과정을 i값을 0으로 초기화하여 i값이 M-1이 될 때까지 반복한 뒤 B[0,M]을 모니터에 출력한다.

## Common prime divisors

임의의 양의 정수로 구성된 숫자열 A[0,M], B[0,M]을 사용자로부터 입력받는다.

또한 output인 counter값을 0으로 초기화한다.

A[i] (0<=i<M) 값을 소인수 분해하여 A[i]에 대한 소인수 집합을 만든다.

B[i]에 대해서도 같은 과정을 진행한다.

그리고 이 두 집합을 비교하여 만약 두 집합이 같다면 counter값을 1 증가시킨다.

두 집합이 다르다면 아무 과정도 진행하지 않는다.

이 과정을 i값을 0으로 초기화하여 i값이 M-1이 될 때까지 반복한 뒤 counter값을 모니터에 출력한다.

## Min abs sum of two

사용자로부터 임의의 정수로 구성된 공집합이 아닌 문자열 A[0,M]을 입력받는다.

(A[p],A[q]) (0<=p<=q<M) 의 합의 절대값들 중 최소인 값을 찾아 출력한다.

## Max pooling

임의의 2N X 2N행렬 A[2N,2N]를 랜덤으로 생성한다.

그리고 N X N행렬 B[N,N]을 생성한 후 모든 element를 0으로 초기화한다.

2X2행렬인 A[i,i+1:j,j+1]에서 각 element들 중 최대값 max를 찾아 B[i,j]값에 대입한다.

이 과정을 for(i=0;i<N;i++):

for(j=0;j<N;j++):만큼 진행 한 뒤 B[N,N]값을 출력한다.

# 모듈 정의

## 모듈 이름: max counter

### 사용자가 처음 입력하는 값 N을 int형태로 저장한다.

그리고 사용자가 입력하는 문자열을 b에 저장한다. 그리고 문자열 method중 split을 이용하여 문자열을 list로 변환하여 A에 저장한다.

output으로 출력할 list c[]를 선언하고 a값만큼 0을 추가한다.

A[i]값을 하나씩 읽어가면서 N값과 비교한다.

A[i]값이 N보다 큰 경우 c[]에서 최대값 max를 찾는다. 그리고 c[]를 max로 초기화시킨다.

A[i]값이 N보다 작은 경우 c[A[i]-1]값을 1만큼 증가시킨다.

모든 A[i]값을 탐색했다면 c를 출력한다.

## 모듈 이름 :Common prime divisiors

### 사용자가 입력한 두 개의 숫자열을 리스트 형태(A,B)로 저장한다.

그리고 각 리스트의 element를 읽어가면서 해당 element의 prime divisors set을 만든다.

A[i],B[i]에 대한 각각의 prime divisors set이 서로 같으면 num값을 증가시킨다.

모든 인덱스를 확인했다면 num값을 출력한다.

## 모듈 이름 : Min abs sum of two

### 사용자가 입력한 숫자열을 A에 저장한다.

그리고 이중 for문을 이용하여 모든 순서쌍에 대해서 합을 구한다.

이때 합이 음수라면 -1을 곱하여 절대값으로 바꿔준다.

모든 순서쌍에 대해서 가장 작은 값을 min에 저장한다.

## 모듈 이름 : Max pooling

### 사용자가 입력한 array의 크기를 num에 저장한다.

그리고 numpy를 이용하여 random한 element를 가지는 num X num array를 만든다.

이번에는 모든 element를 zero로 갖는 절반 크기의 array(arr2)를 만든다.

그리고 2중 for문을 이용하여 2x2행렬 단위로 num X num array를 탐색한다.

각 2x2행렬에서 최대값을 찾아 arr2의 적절한 곳에 대입한다.

# 코드

**1번.**

N = int(input()) # a = N

b = input() # b = string

A =b.split() # A = list of b

c=[] # c = output list

for i in range(N):

c.append(int(0)) #initialize c with 0

for i in range(len(A)):

if int(float(A[i]))>N: # A[i]> N

max = 0

for j in range(N): # find max in c

if c[j]>max:

max = c[j]

for j in range(N): #initialize c with max

c[j]=max

else:

c[int(A[i])-1]+=1 # A[i]<=N --> c[A[i]-1] ++

print(c) # print output

**2번**

A=[]

B =[]

num = 0

primeA=set()

primeB=set()

temp = input()

A= temp.split()

temp = input()

B= temp.split()

for i in range(len(A)):

primeA.clear() #initialize prime divisors set for A

primeB.clear() #initialize prime divisors set for B

for j in range(2,int(A[i])+1): # because 1 could divide every integer and 1 is not prime number. so except 1

if int(A[i])%j==0: #if j could divide A[i] then check whether j is prime number

flag =1

for k in range(1,j):

if j%k==0:

if k!=1: #j is not prime number

flag=0

if flag==1: #j is prime number

primeA.add(j) #add j to prime divisors set for A

for j in range(2,int(B[i])+1):

if int(B[i])%j==0:

flag =1

for k in range(1,j):

if j%k==0:

if k!=1:

flag=0

if flag==1:

primeB.add(j) #to do same thing for B

if primeA.issubset(primeB): # if A <=B

if primeB.issubset(primeA): # and then B<=A then A = B

num+=1

print(num)

**3번**

A = []

temp = input()

A= temp.split()

min = int(float(A[0]))\*int(float(A[0])) # initialize min value why not 0 ? because the abs sum of two should be #equal or greater than 0... so output must be 0 if initial value is 0

for i in range(len(A)):

for j in range(i,len(A)): # check every pair of two element

t= int(A[i])+int(A[j])

if t<0: # if sum is negative

t\*=-1 # than multiple -1 to make it abs sum

if t<=min: #if the sum is smaller than min

min=t #then change the min

print(min)

**4번**

from random import \*

import numpy as np

num = int(input()) #get size from user

k=0

l=0

arr = np.random.random((num,num)) # make num X num random array

arr2 = np.zeros((int(num/2),int(num/2))) # make half size of precious random array and initialize the element zero

for i in range(1,num,2):

for j in range(1,num,2): # to find max value in 2X2 array, use double for loop and make steps 2

max = 0

for k in range(0,2):

for p in range(0,2): # find max value in 2x2 array

if arr[i-k][j-p]>max:

max = arr[i-k][j-p]

arr2[int(i/2)][int(j/2)]=max #and assign the max value to proper position of zero array

print(arr)

print(arr2)