

[참 고]

□ 백준 ATM 문제

문제

인하은행에는 ATM이 1대밖에 없다. 지금 이 ATM앞에 N명의 사람들이 줄을 서있다. 사람은 1번부터 N번까지 번호가 매겨져 있으며, i번 사람이 돈을 인출하는데 걸리는 시간은 P_i 분이다.

사람들이 줄을 서는 순서에 따라서, 돈을 인출하는데 필요한 시간의 합이 달라지게 된다. 예를 들어, 총 5명이 있고, $P_1 = 3, P_2 = 1, P_3 = 4, P_4 = 3, P_5 = 2$ 인 경우를 생각해 보자. [1, 2, 3, 4, 5] 순서로 줄을 선다면, 1번 사람은 3분만에 돈을 뽑을 수 있다. 2번 사람은 1번 사람이 돈을 뽑을 때 까지 기다려야 하기 때문에, $3+1 = 4$ 분이 걸리게 된다. 3번 사람은 1번, 2번 사람이 돈을 뽑을 때까지 기다려야 하기 때문에, 총 $3+1+4 = 8$ 분이 필요하게 된다. 4번 사람은 $3+1+4+3 = 11$ 분, 5번 사람은 $3+1+4+3+2 = 13$ 분이 걸리게 된다. 이 경우에 각 사람이 돈을 인출하는데 필요한 시간의 합은 $3+4+8+11+13 = 39$ 분이 된다.

줄을 [2, 5, 1, 4, 3] 순서로 줄을 서면, 2번 사람은 1분만에, 5번 사람은 $1+2 = 3$ 분, 1번 사람은 $1+2+3 = 6$ 분, 4번 사람은 $1+2+3+3 = 9$ 분, 3번 사람은 $1+2+3+3+4 = 13$ 분이 걸리게 된다. 각 사람이 돈을 인출하는데 필요한 시간의 합은 $1+3+6+9+13 = 32$ 분이다. 이 방법보다 더 필요한 시간의 합을 최소로 만들 수는 없다.

줄을 서 있는 사람의 수 N과 각 사람이 돈을 인출하는데 걸리는 시간 P_i 가 주어졌을 때, 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 사람의 수 N($1 \leq N \leq 1,000$)이 주어진다. 둘째 줄에는 각 사람이 돈을 인출하는데 걸리는 시간 P_i 가 주어진다. ($1 \leq P_i \leq 1,000$)

출력

첫째 줄에 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 출력한다.

예제 입력 1 복사

```
5
3 1 4 3 2
```

예제 출력 1 복사

```
32
```

□ 백준 블랙잭 문제

문제

카지노에서 제일 인기 있는 게임 블랙잭의 규칙은 상당히 쉽다. 카드의 합이 21을 넘지 않는 한도 내에서, 카드의 합을 최대한 크게 만드는 게임이다. 블랙잭은 카지노마다 다양한 규정이 있다.

한국 최고의 블랙잭 고수 김정인은 새로운 블랙잭 규칙을 만들어 상근, 창영이와 게임하려고 한다.

김정인 버전의 블랙잭에서 각 카드에는 양의 정수가 쓰여 있다. 그 다음, 딜러는 N장의 카드를 모두 숫자가 보이도록 바닥에 놓는다. 그런 후에 딜러는 숫자 M을 크게 외친다.

이제 플레이어는 제한된 시간 안에 N장의 카드 중에서 3장의 카드를 골라야 한다. 블랙잭 변형 게임이기 때문에, 플레이어가 고른 카드의 합은 M을 넘지 않으면서 M과 최대한 가깝게 만들어야 한다.

N장의 카드에 써져 있는 숫자가 주어졌을 때, M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 구해 출력하시오.

입력

첫째 줄에 카드의 개수 N($3 \leq N \leq 100$)과 M($10 \leq M \leq 300,000$)이 주어진다. 둘째 줄에는 카드에 쓰여 있는 수가 주어지며, 이 값은 100,000을 넘지 않는 양의 정수이다. 합이 M을 넘지 않는 카드 3장을 찾을 수 있는 경우만 입력으로 주어진다.

출력

첫째 줄에 M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 출력한다.

예제 입력 1 복사

```
5 21
5 6 7 8 9
```

예제 출력 1 복사

```
21
```

알고리즘 스터디 2주차 개인 학습 자료

<포스코 청년 AI·Big Data 아카데미 13기 A반 2조 이경용>

□ 그리디

○ 정 의 : 매 선택에서 할 수 있는 최적의 답을 선택하는 기법. 단, 전체적으로는 최적의 해답이 아닐 가능성 존재

○ 문 제 [참 고]

N명이 ATM 사용을 위해 대기하고 있고 개개인별로 사용 시간이 정해져 있을 때 전체 인원 돈 인출 최소 시간 계산 프로그램 설계

예) 3명의 사용 시간이 1,2,3분일 때 최소 시간: $((1)+(1+2)+(1+2+3)) = 10$ 분

○ 코드 및 해설

코 드	<pre>1 input_value = input("값을 입력해주세요 : ") 2 input_temp = input_value.split() 3 num = int(input_temp[0]) 4 value = list(map(int, input_temp[1:])) 5 sum_value = 0 6 for i in range(num): 7 sum_value += (i+1) * (max(value)) 8 value.remove(max(value)) 9 print("최소값 : ", sum_value)</pre> <p>값을 입력해주세요 : 5 3 1 4 3 2 최소값 : 32</p>
해 설	<p>1번 줄 : 값 입력 인원 수 및 사용 시간 일괄 입력</p> <p>2 ~ 4줄 : 인원, 사용 시간 분리 split 함수 사용하며 분리 및 계산을 위해 int로 전환</p> <p>6 ~ 8줄 : 최소값 계산 최대값 한번, 그 다음 최대값은 두 번 그리고 마지막 값은 인원 수만큼 곱해지고 더해지는 규칙을 따라 작성. 최종 더해진 값은 sum_value에 저장</p>

□ 브루트포스

○ 정 의 : 조합 가능만 모든 경우의 수를 확인하는 알고리즘.

○ 문 제 [참 고]

조합 카드 수 N, 요구되는 카드 합 M이 주어질 때 N개의 카드 중 3개를 골라 합이 M보다 작거나 같은 최대값을 구하는 프로그램 설계

예) 5, 21이 주어졌을 때 5, 6, 7, 8, 9 중 21과 가장 가까운 3장의 카드의 합은 21이다

○ 코드 및 해설

코 드	<pre> 1 input_value = input("값을 입력해주세요: ") 2 input_temp = input_value.split() 3 num = list(map(int, input_temp[:2])) 4 value = list(map(int, input_temp[2:])) 5 sum_result = 0 6 for i in range(num[0]): 7 for j in range(i+1, num[0]): 8 for k in range(j+1, num[0]): 9 sum_temp = value[i] + value[j] + value[k] 10 if (sum_temp <= num[1]) and (sum_temp > sum_result): 11 sum_result = sum_temp 12 print("최대 가까운 값: ", sum_result) 13 </pre> <p>값을 입력해주세요: 10 500 93 181 245 214 315 36 185 138 216 295 최대 가까운 값: 497</p>
해 설	<p>1번 줄 : 값 입력 인원 수 및 사용 시간 일괄 입력</p> <p>2 ~ 4줄 : 인원, 사용 시간 분리 split 함수 사용하며 분리 및 계산을 위해 int로 전환</p> <p>6 ~ 8줄 : 최소값 계산 I는 처음부터, j는 I+1부터, k는 j+1부터 총 카드 수만큼 반복문을 돌려 오늘 경우의 수에 대해서 합 계산 및 최대 근접값 sum_result에 저장</p>