

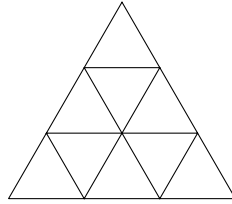
# Rysowanie trójkątów

XVI OIJ, zawody III stopnia  
14 maja 2022

Kod zadania: **tro**  
Limit czasu: **1 s (C++) / 2 s (Python)**  
Limit pamięci: **256 MB**

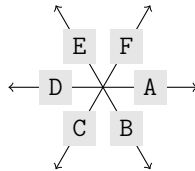


Bajtek postanowił narysować piramidę z trójkątów jak na poniższym rysunku:

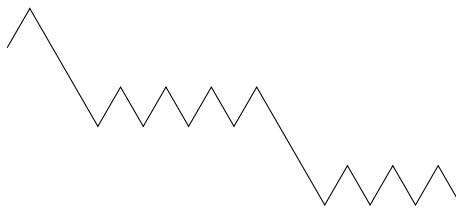


Piramida, którą chce narysować Bajtek składa się z  $N$  pięter. Na najniższym piętrze znajduje się  $2N - 1$  trójkątów umieszczone na przemian czubkiem do góry i czubkiem na dół. Na każdym wyższym piętrze analogicznie umieszczane są trójkąty, ale na każdym poziomie o dwa trójkąty mniej. Powyższy rysunek przedstawia więc sytuację dla  $N = 3$ .

Bajtek ma program, który pozwala mu rysować podobne rysunki na ekranie. W jednym punkcie na ekranie przyłożone jest pióro, które rysuje linie proste pod różnymi kątami. Program Bajtka przyjmuje komendy będące literami A..F – każda litera oznacza przesunięcie pióra w jednym z możliwych kierunków i narysowanie odcinka, zgodnie z poniższym diagramem:



Na przykład, ciąg komend `FBBBBFBFBFBFBFBFBFB` spowoduje narysowanie łamanej takiej jak poniżej:



Nowa wersja programu Bajtka pozwala przyjmować bardziej złożone komendy: oprócz liter możliwe są też cyfry, które oznaczają wielokrotne powtórzenie pewnego ciągu komend. Dokładniej:

- napis postaci  $kZ$ , gdzie  $k \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  oraz  $Z \in \{A, B, C, D, E, F\}$  oznacza to samo, co  $\underbrace{ZZ \dots ZZ}_k$  ( $k$  kopii litery  $Z$ );
- napis postaci  $k[S]$  gdzie  $k \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ , a  $S$  jest pewnym napisem, oznacza  $\underbrace{SS \dots SS}_k$  ( $k$  kopii napisu  $S$ );  
napis  $S$  może przy tym sam zawierać dalsze konstrukcje z cyframi.

Na przykład `3[2A]` to `AAAAA`, a łamaną z powyższego rysunku można więc osiągnąć również krótszymi napisami takimi jak `F3B3[FB]F3B3[FB]`, `2[F3BFBFBFB]` a nawet `2[F3B3[FB]]`. Zwróć uwagę, że napisy, w których  $k$  jest liczbą większą niż 9 (takie jak `10A` lub `58[AB]`) **nie są** dopuszczalne.

Wróćmy do piramidy Bajtka z pierwszego rysunku. Bajtek chce ją zrealizować za pomocą ciągu komend dla swojego programu. Ma dodatkowy warunek: chciałby, żeby pióro nigdy nie rysowało dwukrotnie po tej samej linii (na skutek drobnego buga w programie linia jest wtedy minimalnie grubsza, co drażni poczucie estetyki Bajtka). Przykładowo, napis `2[FB D]` narusza tę zasadę – każda linia trójkąta będzie narysowana dwukrotnie. Drugim wymogiem Bajtka jest, aby program miał nie więcej niż 150 000 znaków.

Bajtek wynajął Ciebie do skonstruowania odpowiedniego ciągu komend. Napisz program, który dla danego  $N$  wypisze ciąg znaków generujący piramidę o wysokości  $N$ . Zakładamy, że pióro na początku znajduje się w lewym dolnym rogu piramidy. Jak to czasem bywa w projektach IT, możesz naruszyć wymagania klienta. To znaczy, że Twój program może dwukrotnie przejść po tej samej linii, albo (trochę) przekroczyć limit 150 000 znaków, ale będzie Cię to kosztowało – w tym wypadku, otrzymasz mniej punktów. Dokładne reguły punktacji podane są w sekcji „Ocenianie”.

## Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna  $N$  ( $1 \leq N \leq 6000$ ), określająca liczbę poziomów piramidy, którą chce narysować Bajtek.

## Wyjście

Twój program powinien wypisać na wyjście wejście do programu Bajtka – napis składający się jedynie ze znaków ze zbioru  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, [, ], A, B, C, D, E, F\}$ . Po wprowadzeniu napisu do programu Bajtka powinna zostać narysowana piramida o  $N$  poziomach, rozpoczynając od lewego dolnego rogu. Długość wypisanego napisu nie może przekraczać 200 000. Aby jednak zdobyć maksymalną liczbę punktów za dany test, długość napisu nie może przekroczyć 150 000 znaków, a każda linia powinna zostać narysowana dokładnie raz.

Zwróć uwagę, że nie jest konieczne wypisanie najkrótszego możliwego napisu, a jedynie zmieszczenie się w tym limicie. Jeżeli istnieje wiele możliwych rozwiązań, Twój program może wypisać dowolne z nich.

## Ocenianie

- Jeżeli Twój napis będzie poprawny i nieznacznie dłuższy niż 150 000 znaków (co najwyżej 200 000 znaków), to wciąż otrzymasz część punktów za daną grupę testów – punktacja w tym przypadku będzie malała liniowo do limitu 200 000 znaków (np. za napis o długości 175 000 otrzymasz 50% punktów). Innymi słowy, jeżeli długość Twojego napisu będzie wynosić  $X$  znaków, to otrzymasz:

$$\begin{array}{ll} 100\% \text{ punktów} & \text{dla } X \leq 150\,000, \\ \frac{200\,000 - X}{500} \% \text{ punktów} & \text{dla } 150\,000 < X \leq 200\,000, \\ 0\% \text{ punktów} & \text{dla } X > 200\,000. \end{array}$$

- Jeżeli Twój program narysuje pewną krawędź wielokrotnie, otrzymasz za daną grupę testów połowę należnej liczby punktów (już po uwzględnieniu punktu powyżej).

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
$N \leq 5$	16
$N \leq 300$	40
$N$ jest parzyste	50
$N$ jest nieparzyste	50

## Przykład

Wejście dla testu tro0:

3

Wyjście dla testu tro0:

3F3BD2E2 [AC] DFDBD



## Pliki

W folderze `/home/zawodnik/rozw` udostępniamy skrypt `narysuj.py` pozwalający na narysowanie rysunku, który odpowiada Twojemu napisowi. Aby go uruchomić, użyj następującej komendy:

```
python3 narysuj.py
```

Następnie możesz podać ciąg znaków (w jednym wierszu), który stanowi wejście do programu Bajtka. Na jego podstawie zostanie narysowana piramida. Twój napis powinien spełniać specyfikację podaną w sekcji "Wyjście". W przeciwnym wypadku nie gwarantujemy poprawnego zachowania programu.

Możesz także uruchomić ten program wraz z napisem zapisanym w pliku. Dla przykładu, aby zobaczyć piramidę generowaną przez napis w pliku `out/tro0.out` użyj następującej komendy:

```
python3 narysuj.py <out/tro0.out
```

Program ten udostępnia też kilka dodatkowych opcji, które możesz poznać wywołując go z flagą `--help`:

```
python3 narysuj.py --help
```