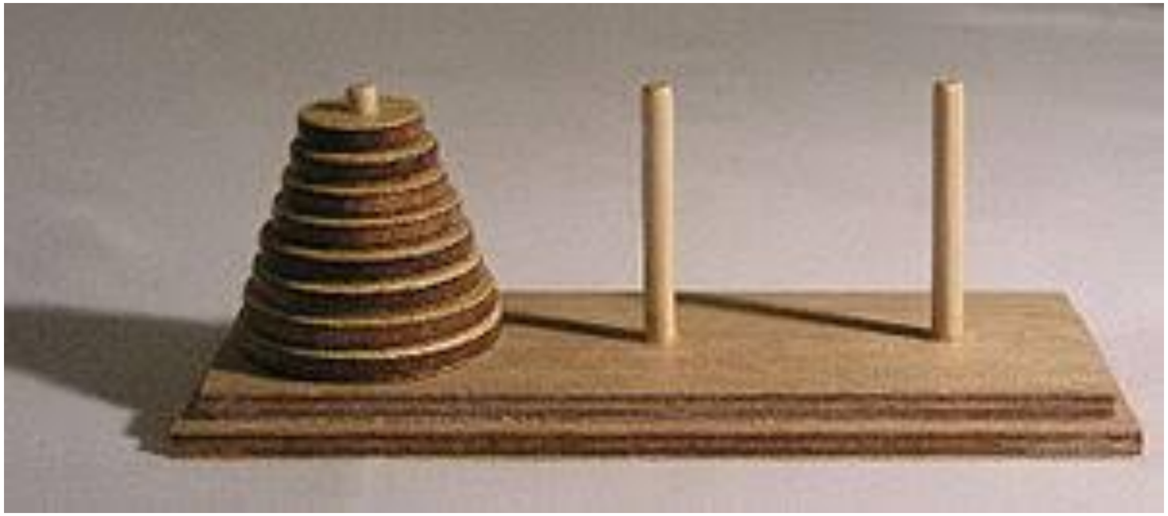
Приложение на стек

доц. д-р. Нора Ангелова

Ханойските кули



https://bg.wikipedia.org/wiki/%D0%A5%D0%B0%D0%BD%D0%BE%D0%B9%D1%81%D0%BA%D0%B0_%D0%BA%D1%83%D0%BB%D0%B0

Ханойските кули

- Реализация с рекурсия

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
    if (k == 1) {
        std::cout << "Move one disk from " << source << " to " << dest << std::endl;
        return;
    }

    towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
    towerOfHanoi(1, source, dest, temp);      // Move 1 disk to dest
    towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
    towerOfHanoi(3, 'A', 'B', 'C');
    return 0;
}
```

Ханойските кули

- Как работи?

```cpp
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
    if (k == 1) {
        std::cout << "Move one disk from " << source << " to " << dest << std::endl;
        return;
    }

    towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
    towerOfHanoi(1, source, dest, temp);     // Move 1 disk to dest
    towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
    towerOfHanoi(3, 'A', 'B', 'C');
    return 0;
}
```

Ханойските кули

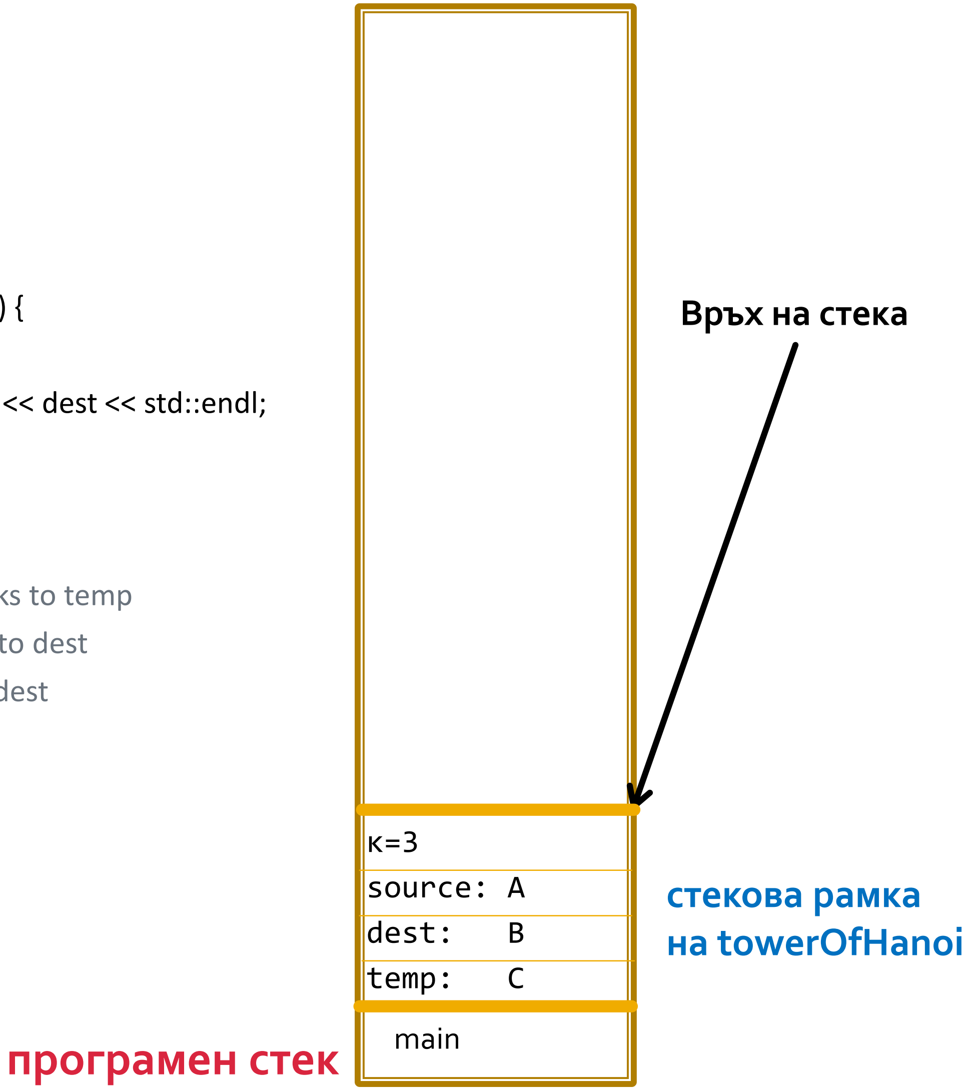
- Как работи?

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
   if (k == 1) {
      std::cout << "Move one disk from " << source << " to " << dest << std::endl;
      return;
   }

   towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
   towerOfHanoi(1, source, dest, temp);       // Move 1 disk to dest
   towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
   towerOfHanoi(3,  'A',  'B',  'C');
   return 0;
}
```



Връх на стека

```
к=3
source:  A
dest:    B
temp:    C
```

стекова рамка на towerOfHanoi

main

програмен стек

Ханойските кули

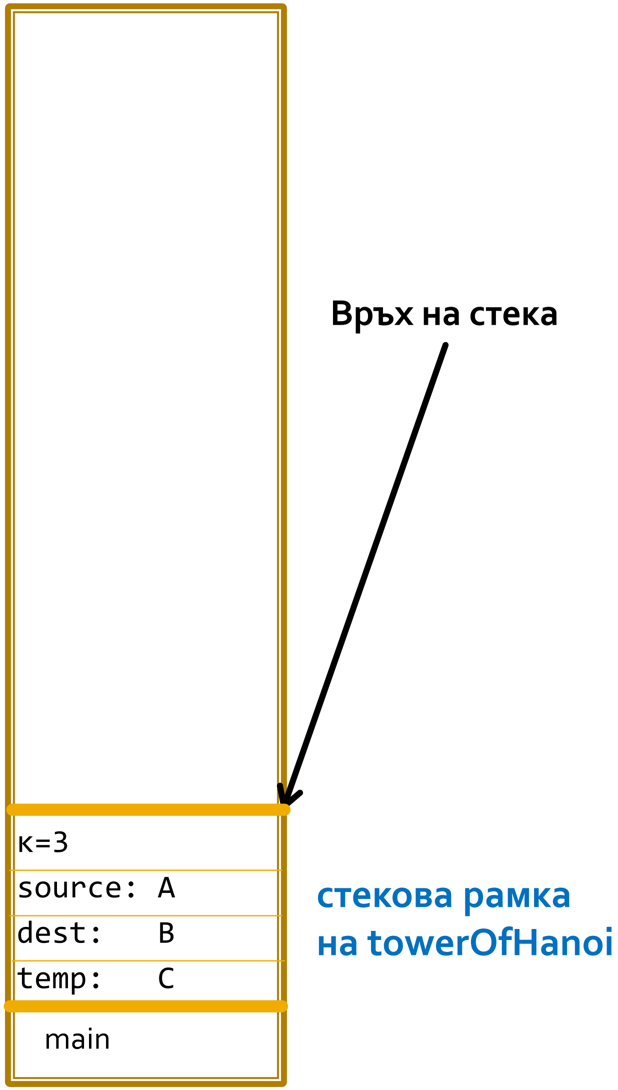
- Как работи?

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
   if (k == 1) {
      std::cout << "Move one disk from " << source << " to " << dest << std::endl;
      return;
   }

   towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
   towerOfHanoi(1, source, dest, temp);       // Move 1 disk to dest
   towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
   towerOfHanoi(3,  'A',  'B',  'C');
   return 0;
}
```



Връх на стека

```
к=3
source:  A
dest:    B
temp:    C
```

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

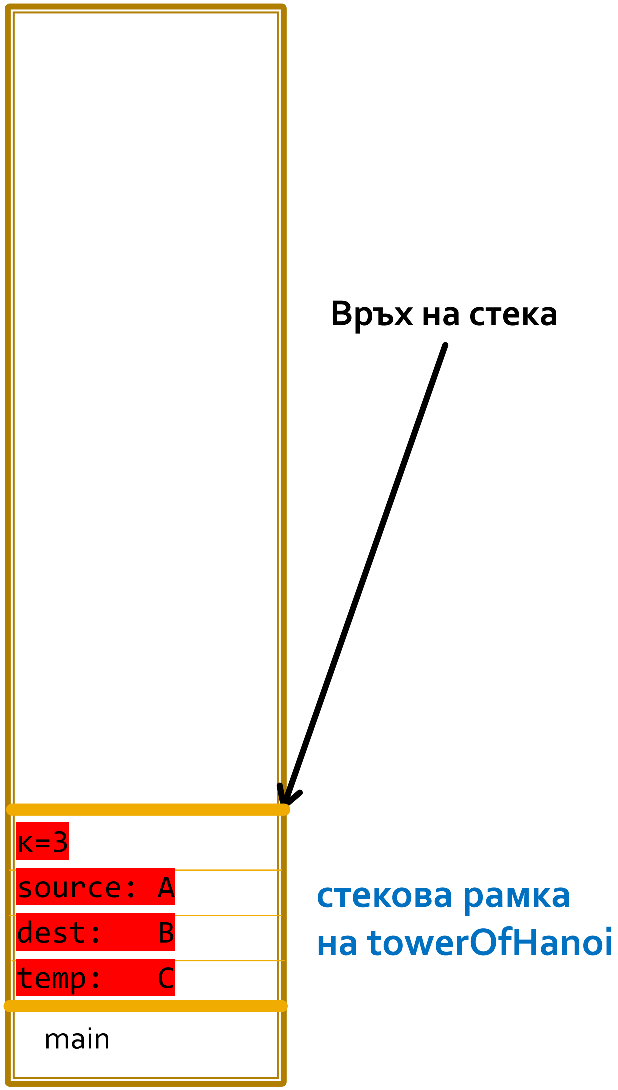
- Как работи?

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
   if (k == 1) {
      std::cout << "Move one disk from " << source << " to " << dest << std::endl;
      return;
   }

   towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
   towerOfHanoi(1, source, dest, temp);      // Move 1 disk to dest
   towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
   towerOfHanoi(3, 'A', 'B', 'C');
   return 0;
}
```



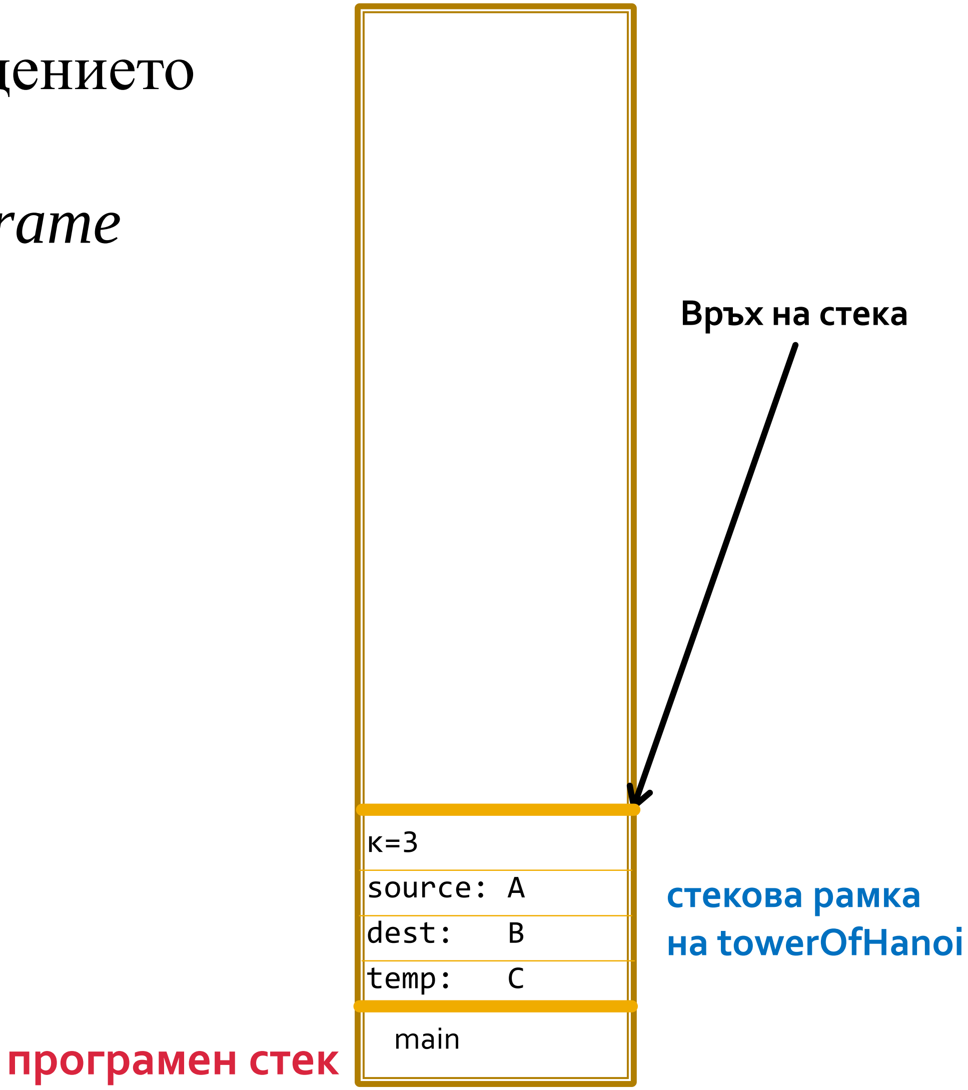
програмен стек

Ханойските кули

- Как да симулираме поведението

Трябват ни рамки - GameFrame

```
struct GameFrame {
  int k;
  char source, dest, temp;
};
```



Връх на стека

к=3

source: A

dest: B

temp: C

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

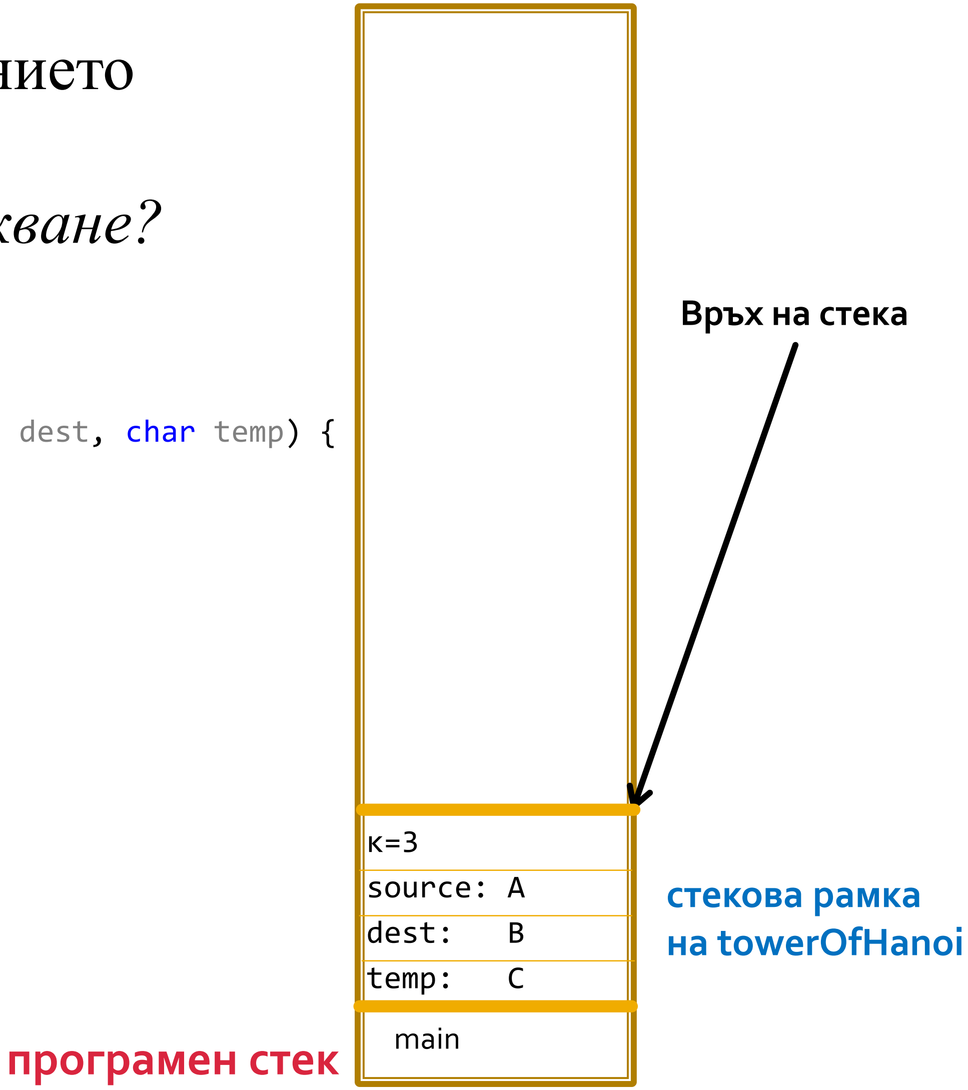
- Как да симулираме поведението

Да симулираме първото извикване?
Трябва ни стек

```
void towerOfHanoiIter(int k, char source, char dest, char temp) {
  std::stack<GameFrame> gameStack;

  gameStack.push({ k, source, dest, temp });

  while (!gameStack.empty()) {
     // …
  }
}
```



Връх на стека

к=3
source: A
dest: B
temp: C

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

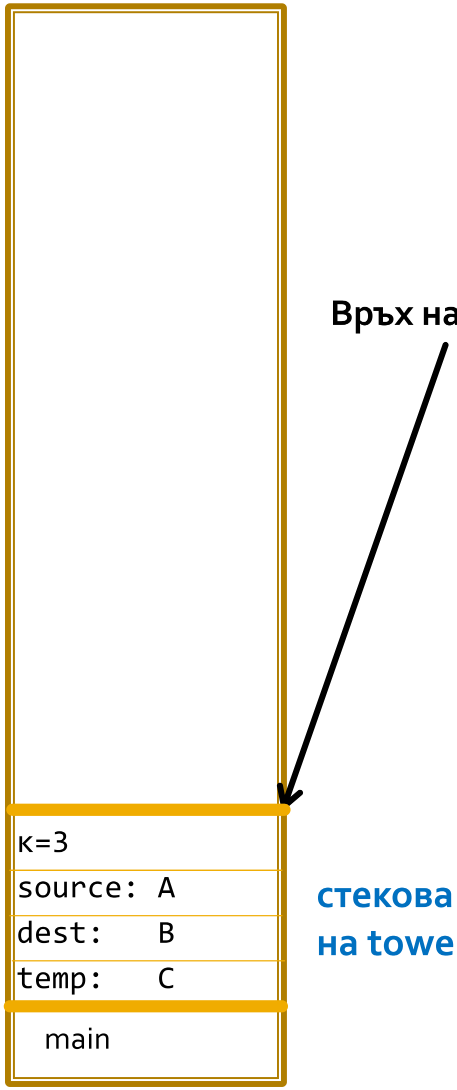
- Как да симулираме поведението

Трябва ни текуща рамка със ст/сти

```
void towerOfHanoiIter(int k, char source, char dest, char temp) {
  std::stack<GameFrame> gameStack;

  gameStack.push({ k, source, dest, temp });

  while (!gameStack.empty()) {
    GameFrame currentFrame = gameStack.top();
    gameStack.pop();
    // …
  }
}
```



Връх на стека

к=3

source: A

dest: B

temp: C

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

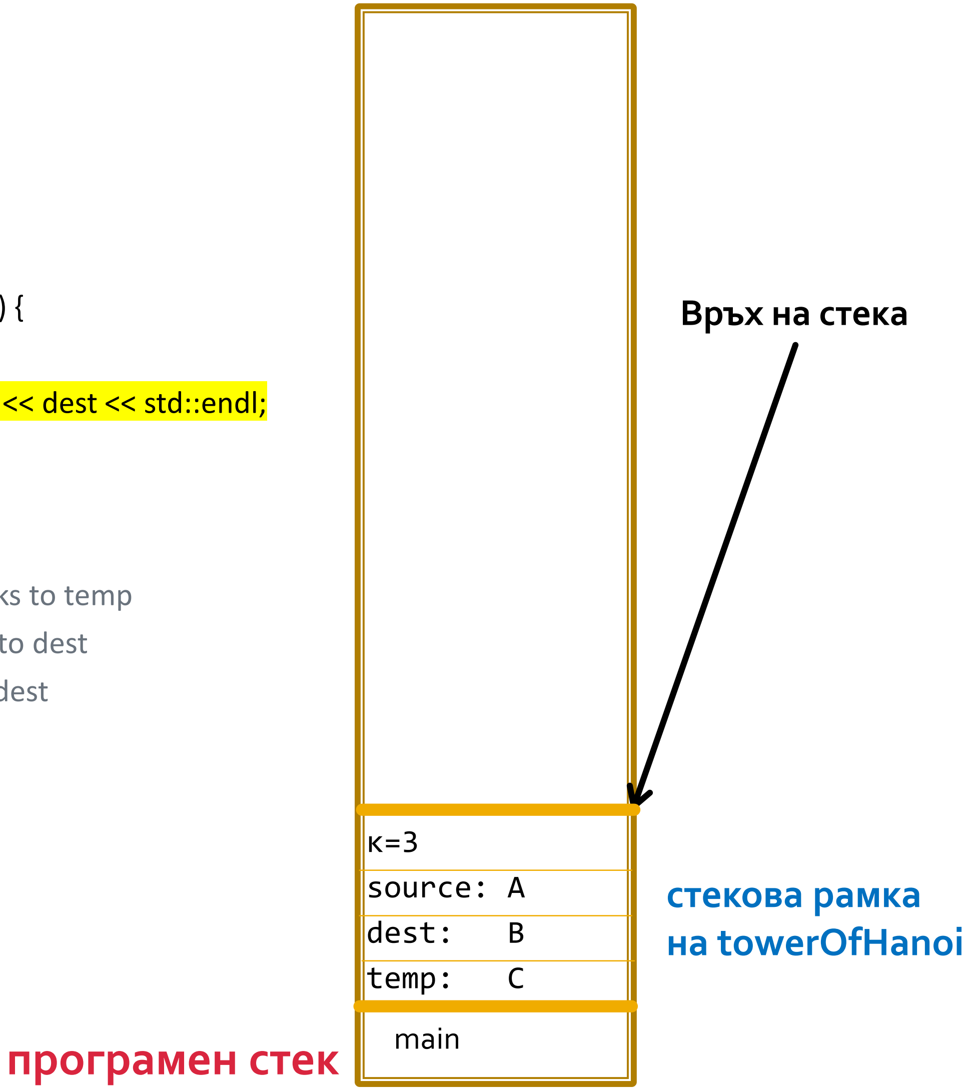
- Как работи?

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
    if (k == 1) {
        std::cout << "Move one disk from " << source << " to " << dest << std::endl;
        return;
    }

    towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
    towerOfHanoi(1, source, dest, temp);       // Move 1 disk to dest
    towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
    towerOfHanoi(3, 'A',  'B',  'C');
    return 0;
}
```



Връх на стека

к=3
source: A
dest: B
temp: C

main

стекова рамка
на towerOfHanoi

програмен стек

Ханойските кули

- Как да симулираме поведението

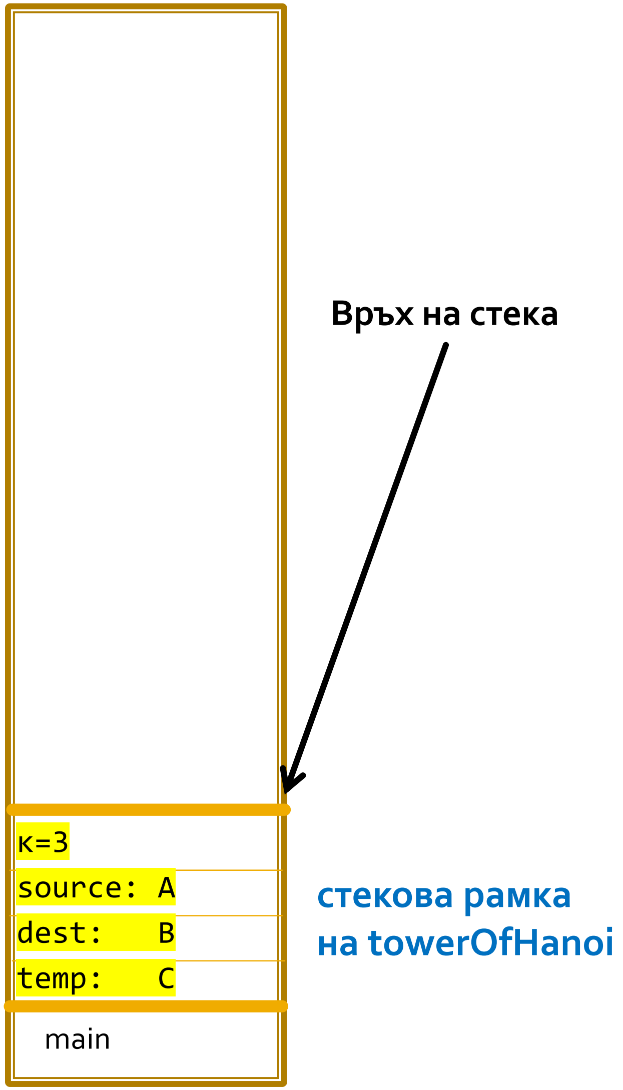
Имаме стойностите
Да симулираме проверката

```
void towerOfHanoiIter(int k, char source, char dest, char temp) {
  std::stack<GameFrame> gameStack;

  gameStack.push({ k, source, dest, temp });

  while (!gameStack.empty()) {
    GameFrame currentFrame = gameStack.top();
    gameStack.pop();

    if (currentFrame.k == 1) {
      std::cout << "Move one disk from " <<
         currentFrame.source << " to " <<
         currentFrame.dest << std::endl;
    }
    // …
  }
}
```



Ханойските кули

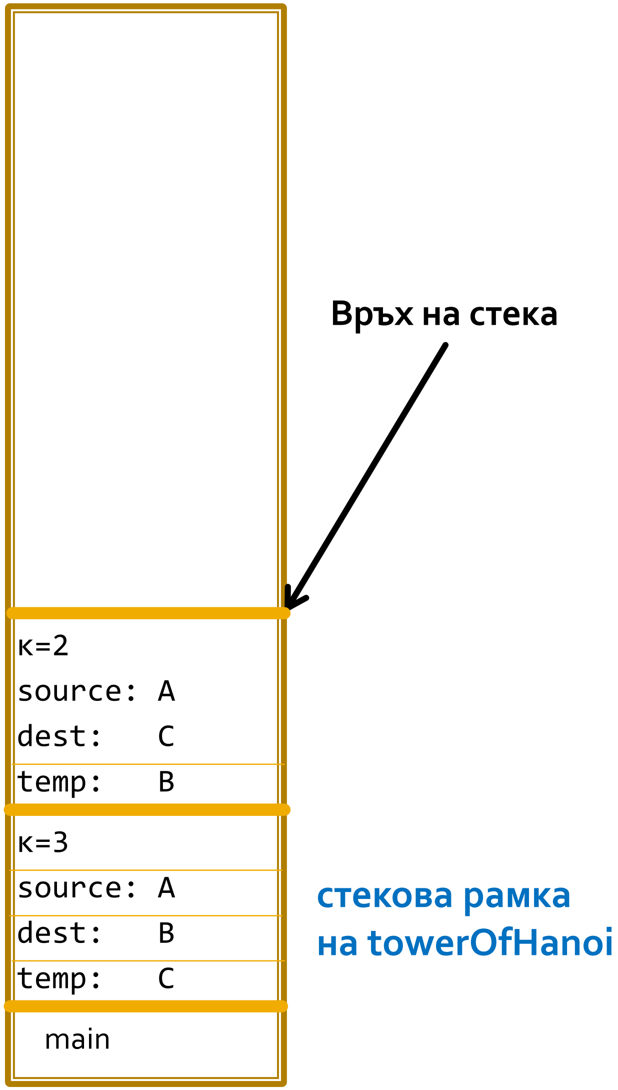
- Как работи?

```
#include <iostream>

void towerOfHanoi(int k, char source, char dest, char temp) {
   if (k == 1) {
      std::cout << "Move one disk from " << source << " to " << dest << std::endl;
      return;
   }

   towerOfHanoi(k - 1, source, temp, dest); // Move k-1 disks to temp
   towerOfHanoi(1, source, dest, temp);     // Move 1 disk to dest
   towerOfHanoi(k - 1, temp, dest, source); // Move k-1 to dest
}

int main() {
   towerOfHanoi(3, 'A', 'B', 'C');
   return 0;
}
```



Връх на стека

к=2
source: A
dest: C
temp: B

к=3
source: A
dest: B
temp: C

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

- Как да симулираме поведението

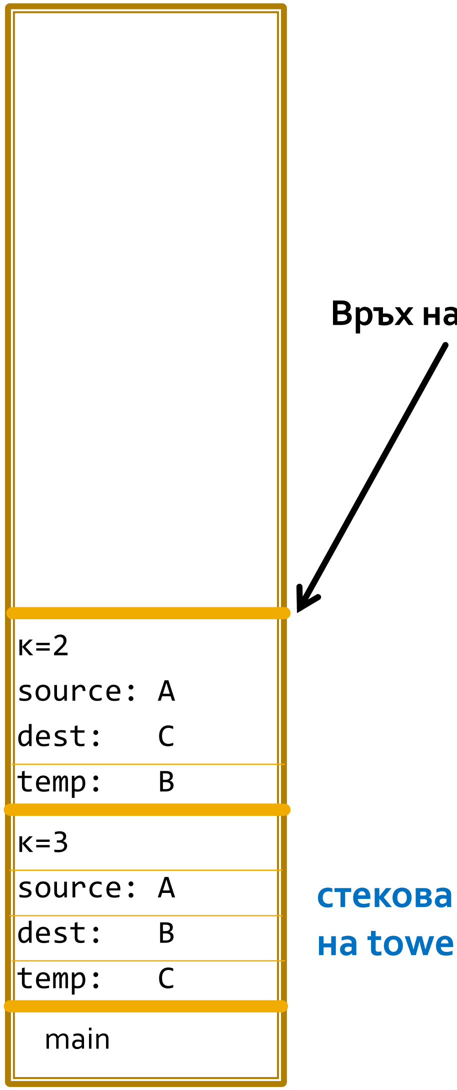
Как да симулираме извикването?
Извикване на ф-я == нова стекова рамка

```
void towerOfHanoiIter(int k, char source, char dest, char temp) {
  std::stack<GameFrame> gameStack;

  gameStack.push({ k, source, dest, temp });

  while (!gameStack.empty()) {
    GameFrame currentFrame = gameStack.top();
    gameStack.pop();

    if (currentFrame.k == 1) {
      std::cout << "Move one disk from " <<
         currentFrame.source << " to " <<
         currentFrame.dest << std::endl;
    } else {
      gameStack.push({
        currentFrame.k - 1,
        currentFrame.temp,
        currentFrame.dest,
        currentFrame.source
      });
      gameStack.push({ … });
      gameStack.push({ … });
    }
  }
}
```



Връх на стека

к=2
source: A
dest: C
temp: B

к=3
source: A
dest: B
temp: C

main

стекова рамка на towerOfHanoi

програмен стек

Ханойските кули

- Как да симулираме поведението

Ред на добавянето на рамките?

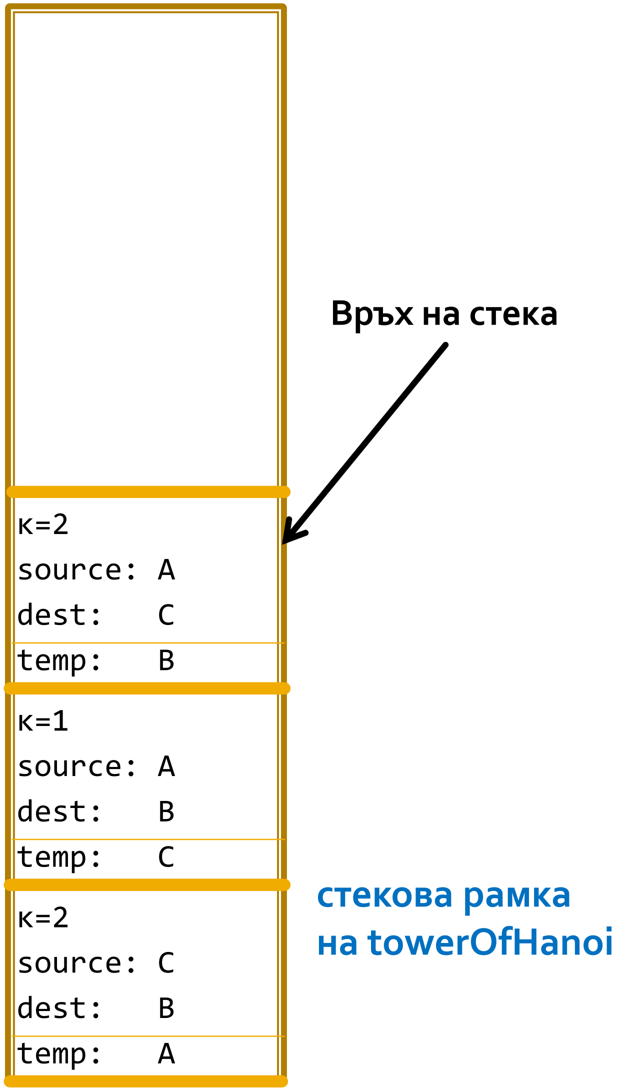
```
void towerOfHanoiIter(int k, char source, char dest, char temp) {
  std::stack<GameFrame> gameStack;

  gameStack.push({ k, source, dest, temp });

  while (!gameStack.empty()) {
    GameFrame currentFrame = gameStack.top();
    gameStack.pop();

    if (currentFrame.k == 1) {
      std::cout << "Move one disk from " <<
         currentFrame.source << " to " <<
         currentFrame.dest << std::endl;
    } else {
      gameStack.push({
        currentFrame.k - 1,
        currentFrame.temp,
        currentFrame.dest,
        currentFrame.source
      });
      gameStack.push({ … });
      gameStack.push({ … });
    }
  }
}
```

GameStack



Връх на стека

```
к=2
source:  A
dest:    C
temp:    B

к=1
source:  A
dest:    B
temp:    C

к=2
source:  C
dest:    B
temp:    A
```

стекова рамка
на towerOfHanoi

Ханойските кули

- Реализация с рекурсия
- Реализация с итерация
- Отделяне на ход

Следва продължение...