

Step 1. C언어의 strchr() 함수

- <string.h> 또는 <cstring> 헤더 필요
- 문자열에서 문자를 검색.
- 문자열의 시작 주소를 받아서 널 문자('\0')를 만날 때까지 검색.
- 구간 종료 조건인 널 문자('\0')도 검색 대상에 포함.
- 검색 실패시 NULL 포인터(0)을 리턴

[개선점]

- 전체 문자열 뿐 아니라 부분 문자열에서 검색가능하게 하면 좋지 않을까 ?
- 종료 조건을 널 문자로 하지 말고 함수 인자로 전달할 수 있도록 변경.

Step 2. 검색 구간의 일반화

1. 구간 종료 조건을 함수 인자로 전달

- 시작 주소와 검색할 요소의 개수를 전달
- 시작 주소와 마지막 요소의 주소를 전달 → STL 이 사용하는 방식

2. 검색 대상

- 시작 주소가 가리키는 곳은 검색에 포함된다.
- 마지막 주소가 가리키는 곳은 검색에 포함되지 않는다.
- 반 개행 구간 (half open range)
- [first, last) 로 표기
- 전체 문자열을 검색하려면 `estrchr(s, s + strlen(s), 'c');`

[개선점]

- 문자열 뿐 아니라 모든 타입의 배열을 검색하는 것이 좋지 않을까 ?

Step 3. 검색 대상 타입의 일반화

1. 함수 템플릿 사용

- 모든 타입의 배열의 [first, last) 구간에서 선형 검색 수행.

2. 함수 이름 변경 : `estrchr()` → `efind()`

[개선점]

- `double`의 배열에서 `int` 검색이 가능하도록 변경.
- 포인터가 아니라도 구간을 표현할 수 있도록 변경.

Step 4. 활용성을 더욱 넓게

1. 이전 단계의 문제점

- 구간의 타입과 찾는 요소의 타입이 연관 되어 있다.(double 배열에서는 double 만 검색 가능, int 검색이 안됨)
- 구간의 표현은 반드시 포인터만 사용해야 한다.

[해결책]

- 구간의 타입과 검색할 요소의 타입을 다른 타입을 사용한다.
- 구간의 시작, 마지막을 전달 받을 때는 포인터(*) 표기를 하지 않는다.

2. 검색 실패의 표현

- 널 포인터(0) 대신 last 리턴.
- 반 개행 구간(half open range) 이므로 last 는 검색에 포함되지 않는다.

Step 5. efind() 함수 정리

1. 검색 대상

- 모든 타입의 배열 의 [first, last) 구간.

2. 검색 구간

- [first, last)
- first 는 검색에 포함 되지만 last는 검색에 포함 안됨(half open range)

3. 구간의 표현

- 포인터 뿐 아니라 객체도 가능. 단, ++, *, !=, == 연산이 가능해야 함.

4. 구간의 이동

- 전위형 증가 연산자.(++)

5. 검색 실패시

- last 리턴

6. C++ 표준 STL 의 find 함수

- <algorithm> 헤더.