

# Undecidability of the Halting Problem

There is no C++ program **halt.cpp** that can decide for every C++ program **P** and every input **I** whether **P** halts on **I**, or runs forever on **I**.

```
// Program halt.cpp:
// test if program P halts on input I
#include<iostream>
#include<string>
```

## Suppose it exists...

```
int main()
{
    std::string P; // program
    std::string I; // input

    std::cin >> P;
    std::cin >> I;
```

```
bool halting;
```

Magic Computation:

P halts on I: `halting` is set to *true*

P runs forever on I: `halting` is set to *false*

```
if (halting)
    std::cout << "yes";
else
    std::cout << "no";

return 0;
```

```
}
```

```
// Program halt.cpp:
// test if program P halts on input I
#include<iostream>
#include<string>
```

`int main()`      Suppose it exists...

```
{
  std::string P; // program
  std::string I; // input

  std::cin >> P;
  std::cin >> I;
```

```
bool halting;
```

Magic Computation:

P halts on I:      `halting` is set to *true*  
P runs forever on I: `halting` is set to *false*

```
if (halting)
  std::cout << "yes";
else
  std::cout << "no";
```

```
return 0;
```

```
}
```

```
// Program selfhalt.cpp:
// test if program P runs forever on itself
#include<iostream>
#include<string>
```

`int main()`      ...then this also exists.

```
{
  std::string P; // program
  std::string I; // input

  std::cin >> P;
  I = P;
```

```
bool halting;
```

Magic Computation:

P halts on I:      `halting` is set to *true*  
P runs forever on I: `halting` is set to *false*

```
if (halting)
  for ( ; ; ) ; // infinite loop
else
  std::cout << "yes";
```

```
return 0;
```

```
}
```

What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>
```

```
int main()    ...then this also exists.  
{
```

```
    std::string P; // program  
    std::string I; // input
```

```
    std::cin >> P;  
    I = P;
```

```
    bool halting;
```

Magic Computation:

P halts on I:            `halting` is set to *true*

P runs forever on I: `halting` is set to *false*

```
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";
```

```
    return 0;
```

```
}
```

What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

If `selfhalt.cpp` halts on `selfhalt.cpp`...

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>
```

```
int main()    ...then this also exists.  
{
```

```
    std::string P; // program  
    std::string I; // input
```

```
    std::cin >> P;  
    I = P;
```

```
    bool halting;
```

Magic Computation:

P halts on I:            `halting` is set to *true*

P runs forever on I: `halting` is set to *false*

```
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";
```

```
    return 0;
```

```
}
```

What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

If `selfhalt.cpp` halts on `selfhalt.cpp`...

...then `selfhalt.cpp` runs forever on `selfhalt.cpp`.

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>
```

```
int main()    ...then this also exists.  
{
```

```
    std::string P; // program  
    std::string I; // input
```

```
    std::cin >> P;  
    I = P;
```

```
    bool halting;
```

Magic Computation:

P halts on I: `halting` is set to *true*

P runs forever on I: `halting` is set to *false*

```
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";
```

```
    return 0;
```

```
}
```

What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

If `selfhalt.cpp` runs forever on `selfhalt.cpp`...

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>  
  
int main()    ...then this also exists.  
{  
    std::string P; // program  
    std::string I; // input  
  
    std::cin >> P;  
    I = P;  
  
    bool halting;  
  
    Magic Computation:  
    P halts on I:      halting is set to true  
    P runs forever on I: halting is set to false  
  
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";  
  
    return 0;  
}
```

What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

If `selfhalt.cpp` runs forever on `selfhalt.cpp`...

...then `selfhalt.cpp` halts on `selfhalt.cpp`.

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>  
  
int main()    ...then this also exists.  
{  
    std::string P; // program  
    std::string I; // input  
  
    std::cin >> P;  
    I = P;  
  
    bool halting;  
  
    Magic Computation:  
    P halts on I:      halting is set to true  
    P runs forever on I: halting is set to false  
  
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";  
  
    return 0;  
}
```



What does `selfhalt.cpp`  
do for  $P = \text{selfhalt.cpp}$ ??

If `selfhalt.cpp` runs forever on `selfhalt.cpp`...

...then `selfhalt.cpp` halts on `selfhalt.cpp`.

```
// Program selfhalt.cpp:  
// test if program P runs forever on itself  
#include<iostream>  
#include<string>  
  
int main() So this doesn't exist!  
{  
    std::string P; // program  
    std::string I; // input  
  
    std::cin >> P;  
    I = P;  
  
    bool halting;  
  
    Magic Computation:  
    P halts on I:      halting is set to true  
    P runs forever on I: halting is set to false  
  
    if (halting)  
        for ( ; ; ) ; // infinite loop  
    else  
        std::cout << "yes";  
  
    return 0;  
}
```

```
// Program halt.cpp:
// test if program P halts on input I
#include<iostream>
#include<string>
```

`int main()` **And neither does this!**

```
{
    std::string P; // program
    std::string I; // input

    std::cin >> P;
    std::cin >> I;
```

```
bool halting;
```

Magic Computation:

P halts on I: `halting` is set to *true*  
P runs forever on I: `halting` is set to *false*

```
if (halting)
    std::cout << "yes";
else
    std::cout << "no";
```

```
return 0;
```

```
}
```

```
// Program selfhalt.cpp:
// test if program P runs forever on itself
#include<iostream>
#include<string>
```

`int main()` **So this doesn't exist!**

```
{
    std::string P; // program
    std::string I; // input

    std::cin >> P;
    I = P;
```

```
bool halting;
```

Magic Computation:

P halts on I: `halting` is set to *true*  
P runs forever on I: `halting` is set to *false*

```
if (halting)
    for ( ; ; ) ; // infinite loop
else
    std::cout << "yes";
```

```
return 0;
```

```
}
```