# Introduction to conditional operator

Look and feel: ? :

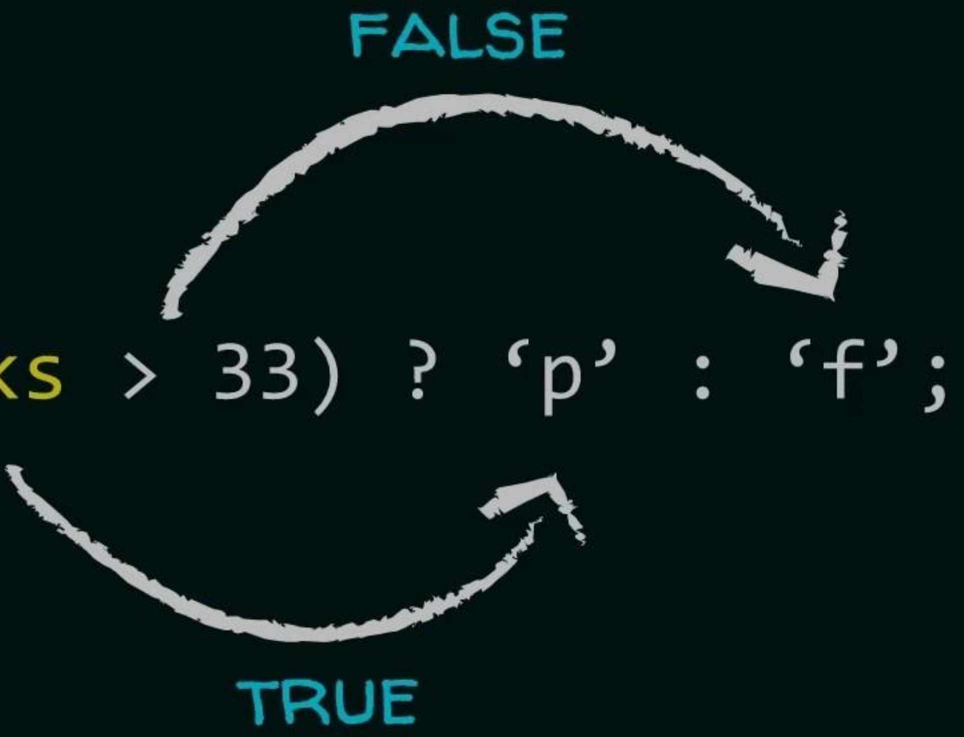
Which one do you choose?

```
char result;
int marks;

if (marks > 33)
{
    result = 'p';
}
else
{
    result = 'f';
}
```

```
char result;
int marks;

result = (marks > 33) ? 'p' : 'f';
```



```
result = (marks > 33) ? ‘p’ : ‘f’;
```

$\boxed{\texttt{(marks > 33)}}$ is a boolean expression, therefore it will return either TRUE or FALSE

$\boxed{\texttt{(marks > 33) ? ‘p’ : ‘f’}}$ is a conditional expression, which is after all an expression, therefore it is an r-value and result is l-value.

## QUICK FACTS CHECKLIST

✓ Conditional operator is the only ternary operator available in the list of operators in C language

✓ As in `Expression1 ? Expression2 : Expression 3`, expression1 is the boolean expression. If we simply write 0 instead of some boolean expression than that simply means FALSE and therefore Expression3 will get evaluated.

Example:

```
int result;
result = 0 ? 2 : 1
```

result

$\boxed{1}$

## Homework problem

What is the output of the following is C program fragment?

```c
#include <stdio.h>

int main() {
    int var = 75;
    int var2 = 56;
    int num;

    num = sizeof(var) ? (var2 > 23 ? ((var == 75) ? 'A' : 0) : 0) : 0;

    printf("%d", num);
    return 0;
}
```