

POLYGLOT PROGRAMMING - AN INTRODUCTION

A **polyglot** is a computer program or script written in a valid form of multiple programming languages, which performs the same operations or output independently of the programming language used to compile or interpret it.

Example

```
# define y >
# include<stdio.h>
# define x z /*
print("shaastra\n");__END__
+++++++[>+++++++>+++++++<<-]
+++[>>+++++<-<-]>><+++++..-----..>+.-.<.* /
main()
{ printf ("shaastra\n"); }
```

This program runs in *C*, *Perl* and *Brainf**k* and prints shaastra.

Usually Polyglots are written as a combination of C (which allows redefinition of tokens with a preprocessor) and a scripting programming language.

First Lets see how to write a polyglot program using macro definition.

Consider the "Hello World" program.

The C code which outputs Hello World is

```
#include<stdio.h>
main(){printf("Hello World\n");}
```

The perl code which outputs Hello World is

```
print("Hello World\n");
or
do{print("Hello World\n");}
```

is a preprocessor statement in C but it is a comment in perl.

using the above fact, a bilingual program which outputs "Hello World" in C and perl can be

```
#include<stdio.h>
#define do main()
#define print printf
do{print("HelloWorld\n");}
```

(or)

```
#include<stdio.h>
#define print(x) main(){printf(x);}
print("Hello World\n")
```

(or)

```
#include<string.h>
#define print(x) main(){puts(x);}
print("Hello World")
```

This technique can be used to write bilingual programs of which one of the language is C and the other is a language like perl, python etc which uses # for defining comments.

An addition polyglot using macro definition...

```
#include<stdlib.h>
#include<stdio.h>
#define raw_input() getchar();getchar();
#define int(x) x;
#define X main(){char a,b,x;int sum=0;x
#define print(x) printf("%d", (int)a+(int)b-96);}
X = 10;
a = raw_input()
a = int(a)
b = raw_input()
b = int(b)
print(a+b)
```

it runs in python and C and adds two single digit numbers.

A three language polyglot which runs in C, python and BrainF**k and add two single digits (if the sum is a single digit)

```
#define i ,>, >+++++[-<----->]<[-<+>]<.>>>>>>
#include<stdlib.h>
#include<stdio.h>
#define raw_input() getchar();getchar();
#define int(x) x;
#define X main(){char a,b,x;int sum=0;x
#define print(x) printf("%d", (int)a+(int)b-96);}
X = 10;
a = raw_input()
a = int(a)
b = raw_input()
b = int(b)
print(a+b)
```

Sometimes using the basic difference between compiled and interpreted languages may be useful...

Example

```
#include<stdio.h>
#define x /*
print("Hai! I am a Perl Code\n");
__END__ */
main()
{printf("Hai! Iam a C Code\n");}
```

Here the Perl Code is commented in C, so it is compatible with C compiler. And the Perl interpreter leaves the first two lines as comments and prints "Hai! I am a Perl Code" and exists from the program.

An addition polyglot in Perl and C which utilises the above fact and macro definition...

```
#include<stdio.h>
#define x /*
$a = <>;
#define y */
#define p /*
$b = <>;
#define q */
#define do main(){int $a,$b;scanf("%d",&$a);scanf("%d",&$b);call($a,$b);}call(int $a,int $b)
#define print(x) printf("%d",x)
do
{print($a+$b);}
```

Polyglot programming is all about utilising the common syntax and different features of different programming languages. Ofcourse it also includes your intelligence.

The following links contains Polyglot programs which you may find useful..

[http://en.wikipedia.org/wiki/Polyglot_\(computing\)](http://en.wikipedia.org/wiki/Polyglot_(computing))
<http://pauillac.inria.fr/~xleroy/stuff/polyglot>
<http://shinh.skr.jp/obf/>
<http://www.retas.de/thomas/computer/programs/useless/misc/polyglot/index.html>
http://nedbatchelder.com/blog/200209/polyglot_programs.html
<http://groups.google.com/group/comp.lang.misc/msg/6167f34c7fff9570>
<http://head.wall.sk/page/polyglot/>
<http://www.gnu.org/fun/jokes/happy-new-year.cfbC.html>