

PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

- In this weeks practical there will be some simple programs which print information to the screen.
- Printing variables is a way to find out what your code is actually doing.
- The function used in C is called `printf`.
- The FORTRAN equivalent is `write`.

- `printf` is part of what are called the standard libraries for C.
- We will be discussing these later in the course.
- However when using `printf` then `stdio.h` must always be included in the program.

int printf(const char* format, ...)

- Your basic formatted output function.
- The first argument is a string containing both static content and special markers marking the formatted output of an evaluated expression.
- The next arguments are a number of expressions that will be evaluated and placed within the static text content of the first argument.

```
int i = 4;
float e = 2.7182818;
double pi = 3.14159265358979;
char c = 'z', s[]="hello world";
printf("i = %d, e = %f, pi = %lf, c = '%c', s = \"%s\"\\n",
      i, e, pi, c, s);
```

printf : format codes

d	int	decimal (base ten) number
o	int	octal number
x or X	int	hexadecimal number
ld	long-sized int	decimal number
u	unsigned	decimal number
lu	unsigned long	decimal number
c	char	single character
s	char pointer	string
f	float	number with 7 digits of precision
g	float	number with up to 7 digits of precision
e	float	number with up to 7 digits of precision, sci.not.
lf	double	number with 16 digits of precision
lg	double	number with up to 16 digits of precision
le	double	number with up to 16 digits of precision, sci.not.

- In FORTRAN output is written to a logical unit. The screen normally has the logical unit number 6.
- A “write” statement is constructed with a set of qualifiers followed by the list of variables. The *r – value* of each variable is output.
- For formatted output:

```
write(UNIT=n, FMT=format, ADVANCE=s, &  
      IOSTAT=int, ERR=label) var1,var2,etc
```

Qualifier	Description
UNIT	logical unit number
FMT	string containing formatting
ADVANCE	'YES' newline (default) or 'NO' no newline
IOSTAT	integer variable, if zero no error
ERR	the program skips to the labeled line, if there is an error

Descriptor	Description
rlw.m	Integer decimal
rBw.m	binary
rOw.m	octal
rZw.m	hexadecimal
	'r' is the number of times repeated
	'w' width of number (including blanks)
	'm' optional minimum of digits (including leading zeros)
rFw.d	real number
rEw.dEe	real number in scientific notation
	'd' is the number of decimals in mantissa
	'e' number characters for exponent
A	character string
rX	space
/	newline

■ Example

```
i = 10; j = 1000000;  
x = 123.456  
  
write(6,fmt='(4x,a,i6,i6,/,5x,a,e10.2e3)',err=10) &  
    ' Test ints ',i,j,' and real ',x  
  
10 continue
```

- The result is (^ indicates a space);

```
^^^^ Test ints      10*****  
^^^^^^ and real 0.12E+03'
```

- The “*****” field is because the integer “j” is too big for the 6 characters provided for it.

- This week we discussed:
 1. hardware architecture (CPU, memory and disc),
 2. language properties and basic layout,
 3. compilation of source code to executable,
 4. memory management through variables.
 5. Printing to screen.