

2021\_\_11\_\_22

November 23, 2021

## 1 2021-11-22

### 1.0.1 Corso ITS

### 1.1 Magento & e-commerce software

### 1.2 ### Fondamenti di Programmazione (Andrea Ribuoli)

### 1.3 Materiale corso

#### 1.3.1 repository: EserciziC presso AndreaRibuoli su *GitHub*

[link](#)

- Verifica delle possibilità di lavoro in **FAD** (*Formazione A Distanza*)
- documento pdf col nome utente GitHub di tutti i colleghi del corso
- sul proprio repository **EserciziC** nel proprio ambiente *GitHub* trasferire una stampa schermo con le informazioni sul sistema operativo del vostro PC
- l'obiettivo minimo è che voi possiate effettuare una iniziale `git clone https://github.com/AndreaRibuoli/EserciziC.git`
- a fronte di successivi aggiornamenti dovete posizionarvi all'interno della cartella ed eseguire una `git pull` (verranno trasferite sole le risorse che hanno subito aggiornamenti)

### 1.4 Tipi di dato elementari

#### 1.4.1 int (*intero*)

```
#include <limits.h>
#include <stdio.h>
int main(void) {
    printf("da %i a %i", INT_MIN, INT_MAX);
    return 0;
}
```

da -2147483648 a 2147483647

In una architettura a 32 bit occupa 4 byte, in una architettura a 64 bit occupa 8 byte.

short <= int <= long

unsigned

unsigned short int [0-65535]

### 1.4.2 char

### 1.4.3 void

#### 1.4.4 float

```
#include <float.h>
#include <stdio.h>
int main(int argc, char *argv[]) {
    float f1, f2;
    double d1, d2, d3;
    f1 = 1000.0;
    f2 = 1000.0000000192092895508;
    d1 = 1000.0;
    d2 = 1000.0000000192092895508;
    d3 = 1000.00000000000000001920;
    printf("da %f a %f\n", FLT_MIN, FLT_MAX);
    printf("da %.19f\n", FLT_EPSILON);
    if (f1 == f2) printf("I float sembrano uguali!\n");
    printf("da %f a %f\n", DBL_MIN, DBL_MAX);
    printf("da %.19f\n", DBL_EPSILON);
    if (d1 == d2) printf("I double d1,d2 sembrano uguali!\n");
    if (d1 == d3) printf("I double d1,d3 sembrano uguali!\n");
    return 0;
}
```

```
[5]: !gcc -o studio_real1 studio_real1.c
```

```
[7]: !studio_reali
```

[illegible]

```
#include <stdio.h>

int main(void) {
    int c = 0;
    while (1) {
        c++;
        printf("Ciao a tutti\n");
        if (c==5) break;
    }
    printf(" voi!");
}
```

```
    return 0;
}
```

```
[8]: !gcc -o esci_da_loop esci_da_loop.c
```

```
[27]: !esci_da_loop
```

```
Ciao a tutti
Ciao a tutti
Ciao a tutti
Ciao a tutti
Ciao a tutti
voi!
```

#### 1.4.5 converti\_api.h

```
#ifndef __CONVERTI_API_H__
#define __CONVERTI_API_H__
```

```
int converti_api(char *CP, char *utf8);
```

```
#endif
```

#### 1.4.6 converti\_api.c

```
/*
```

```
Autore:                      Andrea Ribuoli (2021)
nell'ambito di:             FONDAMENTI DI PROGRAMMAZIONE
                           ITS "Magento & e-commerce software"
```

```
*/
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include "converti_api.h"
```

```
static union { char c[8]; unsigned long a_long; } test = { { 'l', '?', '?', '?', '?', '?', '?', '?' }
#define ENDIANNESS ((char) test.a_long)
```

```
void inverti(char *s);
```

```
int converti_api(char *CP, char *utf8) {
    char *end;
    char str[9] = { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    unsigned long *b = (unsigned long *) (str);
    unsigned long i;
```

```

i = strtol(CP, &end, 16);
if (i <= 127) {
    *b = i;
    if (sizeof(long)==8) {
        if (ENDIANNESS == 'l') inverti(str);
        sprintf(utf8, "%s", &(str[sizeof(long)-1]));
    }
} else if (i <= 2047) {
    *b = (((i & 0x07C0) << 2) | (i & 0x003F)) | 0xC080;
    if (sizeof(long)==8) {
        if (ENDIANNESS == 'l') inverti(str);
        sprintf(utf8, "%s", &(str[sizeof(long)-2]));
    }
} else if (i <= 65535) {
    *b = (((i & 0xF000) << 4) | ((i & 0x0FC0) << 2) | (i & 0x003F)) | 0xE08080;
    if (sizeof(long)==8) {
        if (ENDIANNESS == 'l') inverti(str);
        sprintf(utf8, "%s", &(str[sizeof(long)-3]));
    }
} else {
    *b = (((i & 0x03F000) << 4) | ((i & 0x0FC0) << 2) | (i & 0x003F)) | 0xF0808080;
    if (sizeof(long)==8) {
        if (ENDIANNESS == 'l') inverti(str);
        sprintf(utf8, "%s", &(str[sizeof(long)-4]));
    }
}
return 0;
}

void inverti(char *s) {
    char p;
    p = s[0]; s[0] = s[7]; s[7] = p;
    p = s[1]; s[1] = s[6]; s[6] = p;
    p = s[2]; s[2] = s[5]; s[5] = p;
    p = s[3]; s[3] = s[4]; s[4] = p;
}

```

#### 1.4.7 test\_driver.c

```

#include <stdio.h>
#include <stdlib.h>

#include "converti_api.h"

int main(int argc, char *argv[]) {
    char risultato[5];
    for (int c = 1; c < argc; c++) {
        converti_api(argv[c], risultato);
    }
}

```

```

    printf("%s (UTF8:%s)\n", argv[c], risultato);
}
return 0;
}

```

```
[32]: !gcc -c -o converti_api.o converti_api.c; \
gcc -c -o test_driver_api.o test_driver_api.c; \
gcc -o tester converti_api.o test_driver_api.o
```

```
[34]: !tester 41 42 61 62 3C0 20AC
```

```

41 (UTF8:A)
42 (UTF8:B)
61 (UTF8:a)
62 (UTF8:b)
3C0 (UTF8: )
20AC (UTF8:€)

```

#### 1.4.8 tabella\_ascii.c

```

#include <stdio.h>
#include <stdlib.h>

#include "converti_api.h"

int main(int argc, char *argv[]) {
    char risultato[5];
    printf(" ");
    for (int c = 0; c < 16; c++) printf("| -%X ", c);
    printf("\n");
    for (int c = 0; c < 17; c++) printf("|----", c);
    printf("\n");
    for (int r = 2; r < 8; r++) {
        printf("| %X- ", r);
        for (int c = 0; c < 16; c++) {
            char ascii[3];
            sprintf(ascii, "%2.2X", 16 * r + c);
            converti_api(ascii, risultato);
            printf("| %s ", risultato);
        }
        printf("\n");
    }
    return 0;
}

```

```
[1]: !gcc -c -o tabella_ascii.o tabella_ascii.c; \
gcc -o gen_ascii converti_api.o tabella_ascii.o
```

```
[2]: !./gen_ascii
```

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
2-		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	