

Tabella ridotta codici ASCII - es. 'A' ->0x41 -> 4*16+1 -> 65 ->0100 0001

Hex	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	{		}	~	□

In tutti gli esercizi sotto riportati, per i valori di variabili non definite nel testo dell'esercizio, fare riferimento al riquadro valori presente sul foglio risposte.

Ogni esercizio usa le variabili con i valori definiti e inizializzati – NON si tenga conto di eventuali cambiamenti di valore tra esercizio e l'altro.

1. Conversioni di formato. [p.1]

Completare la tabella riportata al punto 1 del foglio risposte effettuando le conversioni richieste facendo uso dei valori riportati nel riquadro valori.

2. Espressioni aritmetiche - [p.1] Determinare il valore assunto da VAR dopo la istruzione

`VAR = S.CO[2] + B / 015 - V + S.Index;`

dove VAR è una variabile di tipo double

3. Operatori bitwise, shift e manipolazione bitmask - [p.3] Determinare il valore della espressione

`(V<<1 | 7>>1 & 0xB)`

e scriverne la corretta rappresentazione in decimale. *Suggerimento: può essere utile aiutarsi con la griglia*

4. Date le seguenti istruzioni indicare il valore stampato dalle istruzioni seguenti: [p.2]

`x = S.CO[5] << 1; x = x >> 1; y = ~x; printf("%d",y);`

5. Operatori aritmetici [p.2] - dire qual è l'output che si ottiene dalla esecuzione delle seguenti frasi.

```
int w=A;
printf("%d ", w++);
printf("%d ", w);
printf("%d ", ++w);
printf("%d ", w+1);
printf("%d ", --w);
printf("%d ", w);
printf("%d ", w++);
```

6. Strutture condizionali [p.3] - Stabilire qual è l'output che si osserva.

```
int w=A;
if(w%2 == 0)
{
    if(w*w <= 49)
    {
        if(++w == 5) printf("%d",w+V);
        else printf("%d",w-V);
    }
    else if ( w++ < 13) printf("%d", w);
    else printf("%d", w);
}
else if(w%3 == 0)
{
    if(w<10)printf("%d",V);
    else printf("%d",w*10);
}
else printf("%d",w+V);
```

7. **Combinazione di costrutti di controllo [p.3]** - determinare il valore della variabile x a fine esecuzione.

```
main()
{int i, x=0;
  i = S.MATR/S.PAR;
  if ((i<S.CO[5]) && (i > S.NO[5] - S.CO[4]))x++;
  for( ; i>0; i--) x++;
  while( S.PAR % 2 == 0){
    x+=10;
    break;
  }
}
```

8. **Espressioni logiche [p.3]** - Determinare l'output del seguente frammento di programma.

```
int b=B;
printf("%d ", (A<=20 || A>=40));
printf("%d ", (! (b=A*2)));
printf("%d ", (A<=B && A<=V));
printf("%d ", (A>=B && A>=V));
printf("%d ", (! (A>=B && A<=V)));
printf("%d ", (! (A>=B) || ! (A<=V)));
```

9. **Formati di I/O [p.1]** - Che cosa stampa questa istruzione?

```
printf("%4.2f %d %c%c", S.Index, S.PAR, S.CO[2], S.NO[1]);
```

10. **Costrutti condizionali [p.3]** - Che cosa stampa il seguente frammento di programma?

```
int i;
i=(S.CO[4]& 0x0F) >> 1;
if((i&1)==1)
{
  switch(i++)
  {
    case 3:printf("1010");
    case 5:printf("0011");
    default:printf("0001");
  }
}
else
{
  switch(++i)
  {
    case 3:printf("0101");
    case 5:printf("1100");
    default:printf("1110");
  }
}
```

11. **Funzioni ed espressioni logiche [p.3]** - Assumendo la funzione F1 definita come segue,

```
int F1( char X )
{ if (X<= 'C') return 1;
  else if (X>= S.CO[3]) return 2;
  else if (X== 'Y') return 3;
  else if (X<= S.NO[2]) return 4;
  else return 5; }
```

che cosa stampa la seguente istruzione?

```
printf("%d", F1(S.CO[5]));
```