

# MANUALE D'USO

## UTENZE ELETTRICHE

Guida all'utilizzo

[0.00/23.00kW]		
[A]	Aspirapolvere	4.00
[B]	Lavatrice	8.00
[C]	Televisione	6.00
[D]	Phon	4.00
[E]	Telefono	6.00

# INDICE

<b>Interfaccia Grafica</b> .....	3
Potenza Utilizzata e Totale.....	3
Utenze.....	4
Gestione Utenze.....	5
Dimensioni Finestra.....	6
Implementazione.....	7
<b>Manutenzione</b> .....	8
Struttura Immissione Dati.....	8
<b>Algoritmi</b> .....	10
Criteri.....	10
Priorità.....	11
Attivazione.....	13
Disattivazione.....	13

# INTERFACCIA GRAFICA

## POTENZA UTILIZZATA E TOTALE

Nella parte superiore dell'interfaccia grafica, è possibile notare due valori separati da uno slash (/). Il valore sulla sinistra è contrassegnato in verde, rappresenta la **potenza utilizzata**, il valore sulla destra invece rappresenta la **potenza massima** supportata dal programma ed è scritto in giallo.



# INTERFACCIA GRAFICA

## UTENZE

Il corpo dell'interfaccia grafica è composto da tante righe quante utenze sono presenti all'interno del programma. All'inizio di ogni riga è presente una lettera tra parentesi quadre: indica il **pulsante** da premere per attivare e disattivare l'utenza. Vi è successivamente il **nome** dell'utenza, seguito da un valore che indica la **potenza** che occupa il dispositivo. Infine vi è un segnale che indica lo **stato attuale** dell'indicatore: verde acceso, rosso spento.

[8.00/23.00kW]			
[A]	Aspirapolvere	4.00	■
[B]	Lavatrice	8.00	■
[C]	Televisione	6.00	■
[D]	Phon	4.00	■
[E]	Telefono	6.00	■

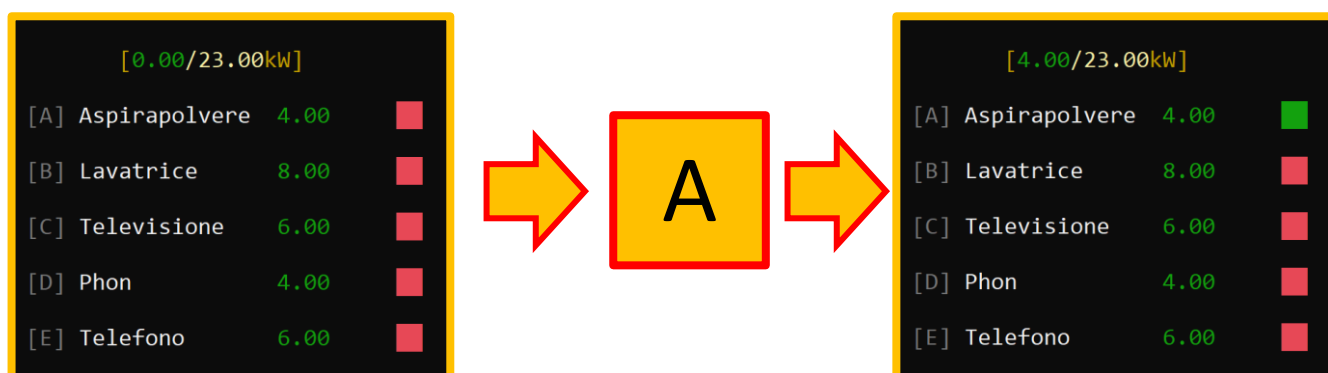
ON

OFF

# INTERFACCIA GRAFICA

## GESTIONE UTENZE

Per l'attivazione e la disattivazione di ogni utenza, è necessario premere il tasto associato, che è possibile vedere sulla sinistra del nome dell'utenza interessata. Al cliccare della stessa, il contatore della potenza utilizzata aumenterà di quanto specificato a destra del nome se spenta, altrimenti diminuirà.

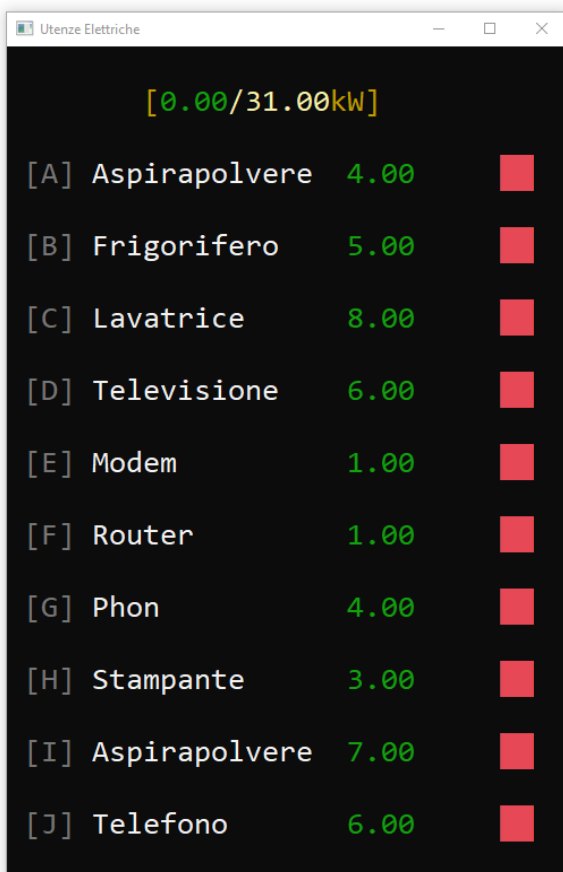


Se la potenza utilizzata raggiunge il limite, il programma risponde **variando lo stato di alcune utenze**, secondo la loro priorità. Tale aspetto viene approfondito andando avanti con il manuale.

# INTERFACCIA GRAFICA

## DIMENSIONI FINESTRA

Le dimensioni della finestra, variano in base al numero di utenze. Ad ogni utenza aggiunta, il programma aggiungerà due righe di altezza al pannello: una che ospita la nuova utenza e una che la separa dalla precedente.

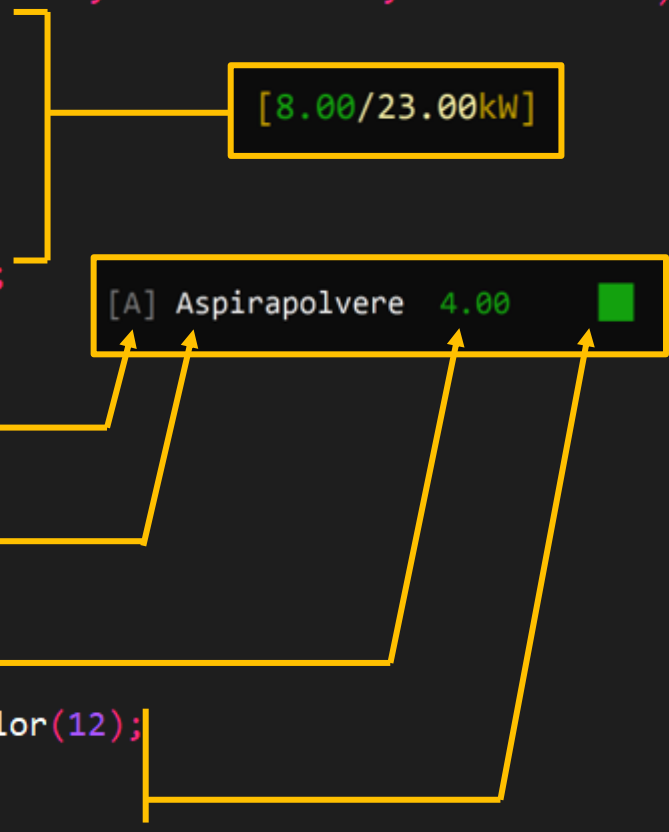


```
void windowSize(int arrSize) {  
    char windowMessage[30];  
    sprintf(windowMessage, "mode con: cols=%d lines=%d", 33, 3 + arrSize*2);  
    system(windowMessage);  
}
```

# INTERFACCIA GRAFICA

## IMPLEMENTAZIONE

```
void InterfacciaGrafica(Utenza* arr, int arrSize, float arrUsed, float arrMAX){
    gotoxy(9, 2);
    textcolor(6); printf("[");
    textcolor(2);
    printf("%.2f", arrUsed);
    textcolor(14); printf("/%.2f", arrMAX);
    textcolor(6); printf("kW]");
    for(int i = 0; i < 10; i++) printf(" ");
    for(int i = 0; i < arrSize; i++) {
        textcolor(8);
        gotoxy(2, i*2 + 4);
        printf("[%c]", 'A'+i);
        textcolor(15);
        gotoxy(6, i*2 + 4);
        printf("%s", arr[i].Nome);
        textcolor(2);
        gotoxy(6 + nameSize, i*2 + 4);
        printf("%3.2f", arr[i].Potenza);
        arr[i].isOn ? textcolor(2) : textcolor(12);
        gotoxy(15 + nameSize, i*2 + 4);
        printf("%c%c", 219, 219);
    }
}
```



Parametri inviati:

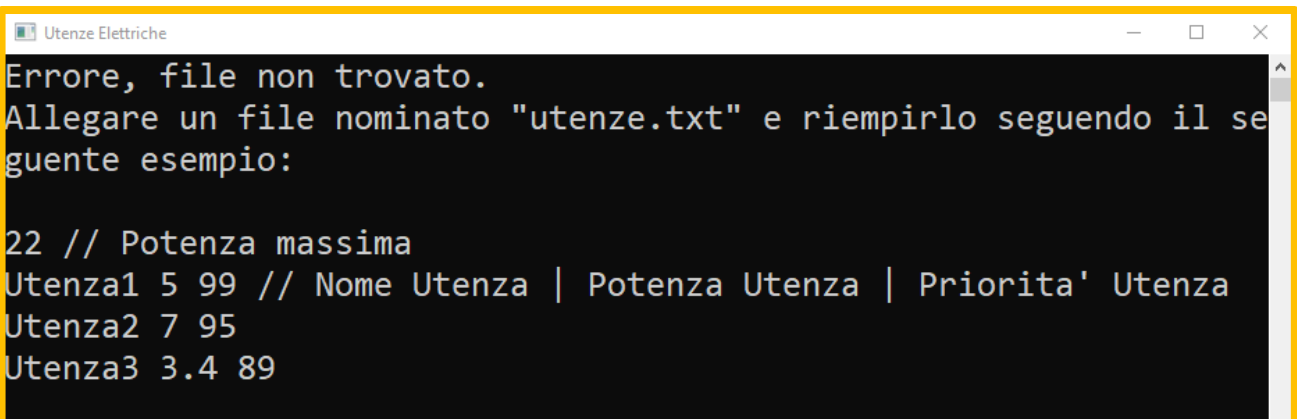
- **arr:** Struttura dati contenente tutte le informazioni relative alle utenze.
- **arrSize:** Valore numerico contenente il numero di utenze presenti all'interno del sistema.
- **arrUsed:** Potenza utilizzata: la somma di tutte le potenze delle utenze attive.
- **arrMAX:** Potenza Massima supportata dal sistema utilizzato.

# MANUTENZIONE

## STRUTTURA IMMISSIONE DATI

Per variare il limite della potenza, è necessario conoscere prima il metodo utilizzato dal programma per l'immissione di utenze e limite potenza. All'avvio del programma, il file cercherà nella stessa cartella in cui si trova, un file denominato **utenze.txt**.

Qualora il file non esistesse, il programma risponderebbe segnalando all'utente l'errore appena riscontrato:



```
Utenze Elettriche
Errore, file non trovato.
Allegare un file nominato "utenze.txt" e riempirlo seguendo il seguente esempio:

22 // Potenza massima
Utenza1 5 99 // Nome Utenza | Potenza Utenza | Priorita' Utenza
Utenza2 7 95
Utenza3 3.4 89
```

Il file di input, per essere correttamente interpretato, deve rispettare la seguente sintassi:



```
utenze.txt - Blocco note di Windows
File Modifica Formato Visualizza ?
23
Aspirapolvere 4 9999
Lavatrice 8 999
Televisione 6 99
Telefono 6 1
Phon 4 9
```

La prima riga è riservata a un valore di natura intera: rappresenta il limite della potenza massima.

Le successive righe invece, si occupano di contenere informazioni relative alle utenze, ogni riga un utenza differente.

Per ogni utenza, è necessario specificare le seguenti informazioni, separate da uno spazio:

- Nome dell'utenza, senza nessuno spazio di separazione.
- La potenza dell'utenza, quindi il consumo provocato dall'accensione della stessa. I numeri reali sono accettati.
- La priorità, più il numero è elevato, più c'è possibilità che l'utenza non venga spenta. I valori inseribili sono di natura intera.

# ALGORITMI

## CRITERI

Le utenze vengono disattivate e attivate, qualora il limite della potenza venga superato. I criteri per stabilire la disattivazione di un'utenza rispetto a un'altra vengono definiti a seconda della priorità specificata. L'algoritmo che ne permette il corretto funzionamento si divide in due:

- Disattivazione:

Nella parte di disattivazione, il programma si occupa solamente di disattivare le utenze finchè non si è giunti a una situazione dove la potenza utilizzata è inferiore rispetto a quella totale.

- Attivazione:

Nella seconda parte invece, avviene un vero e proprio “recupero”. Tutto ciò che è stato disattivato nel processo di prima, non è detto che sia necessario che rimanga spento. A questo punto avviene una riattivazione di utenze che sono state disattivate inutilmente. Il processo ha senso che venga inizializzato se e solo la potenza limite non viene superata da quella utilizzata.

# ALGORITMO

## PRIORITÀ

Durante la prima fase, le utenze vengono disattivate dalla meno importante alla più importante, per ovvie ragioni. La struttura dati viene quindi ordinata, in modo da permettere una facile implementazione del programma. L'algoritmo di ordinamento scelto è l'InsertSort, il migliore in questo caso che potessimo scegliere.

```
bool InsertSort(Utenza* arr, int &arrSize, Utenza temp){
    if( arrSize < 0 || arrSize >= MAX) return false;
    int f = arrSize - 1;
    // "Order by ..., then ..."
    // Ordinare secondo X, poi secondo Y:
    // X: Priorità dell'utenza.
    // Y: Potenza inferiore.
    float arrComparison = arr[f].Priorita + 1.0 - arr[f].Potenza / pow(10,6);
    float tempComparison = temp.Priorita + 1.0 - temp.Potenza / pow(10,6);

    while( f >= 0 && arrComparison < tempComparison){
        arr[f + 1] = arr[f];
        f--;
        arrComparison = arr[f].Priorita + 1.0 - arr[f].Potenza / pow(10,6);
    }

    arr[f + 1] = temp; arrSize++;
    return true;
}
```

Pian piano che le utenze vengono lette dal file, vengono ordinate. Le righe che sicuramente potrebbe risultare confusionarie sono le seguenti.

L'utilità di stabilire tali assegnamenti è di ordinamento. Qualora ci siano utenze con lo stesso valore come priorità, il programma

assegnerà una priorità maggiore alle utenze a potenza inferiore.

Per **arrComparison** si intende una variabile che contiene il nuovo valore contenente priorità e potenza(in dimensioni nettamente inferiore). Ad ogni ciclo cambia valore, in quanto nella struttura sono presenti molteplici utenze.

Per **tempComparison** si intende una variabile che contiene il nuovo valore contenente priorità e potenza(in dimensioni nettamente inferiore). Si riferisce all'utenza nuova, quindi quella da inserire all'interno della struttura.

*float tempComparison =*

*temp.Priorita + 1.0 - temp.Potenza/pow(10,6)*

“Al valore della Priorità, viene aggiunto una minima parte della potenza(valore sotto lo zero), che quindi è rilevante solo se le due priorità sono uguali.”

# ALGORITMO

## ATTIVAZIONE

Durante il processo di Attivazione, l'array viene scorso dal basso verso l'alto, quindi dall'elemento meno importante al più importante.

La funzione **AggiornaUtenza(...)** cambia il valore dell'utenza specificata: se attiva la spegne e viceversa.

```
bool isIdle[MAX];

int i = arrSize - 1;
while( i >= 0 && arrUsed > arrMAX ) {

    if( arr[i].isOn && i != fixed ) {

        AggiornaUtenza(arr, arrSize, i, arrUsed, arrMAX);
        isIdle[i] = true;

    } else isIdle[i] = false;

    i--;
}

// Gli elementi restanti, hanno valore isIdle = false
while( i >= 0 ) isIdle[i--] = false;
```

Variabili utilizzate:

- **isIdle**: Array di bit, indicano gli elementi disattivati.
- **arrUsed**: Potenza utilizzata.
- **arrMAX**: Potenza limite.

- **arr**: Struttura Dati contenente informazioni inerenti alle utenze.
- **fixed**: Posizione dell'ultimo elemento inserito all'array.

Man mano che l'array scorre, vengono memorizzati gli elementi disattivati, saranno utili nella seconda parte. Inoltre l'ultimo elemento inserito non viene disattivato. L'array verrà scorso finchè non finirà, oppure finchè la potenza utilizzata non supera quella massima tollerata.

## DISATTIVAZIONE

```
i = 0;
while( i < arrSize ) {
    if( isIdle[i] && arr[i].Potenza + arrUsed <= arrMAX ) {
        AggiornaUtenza(arr, arrSize, i, arrUsed, arrMAX);
    }
    i++;
}
```

Nella fase due, gli elementi che sono stati disattivati, se possono essere attivati senza superare il limite, allora verranno attivati. Sta volta si parte dall'elemento più importante, quindi dall'alto verso il basso. Verranno abilitate solo le utenze disattivate in precedenza.