

Politecnico di Milano - Anno Accademico 2005-06 - Informatica C
Appello 10 luglio 2006 – A

COGNOME e NOME

Matricola

Es 1. [punti 5]: Files

Scrivere una funzione che apre in lettura il file input.txt e in scrittura il file output.txt (creandolo se non esiste già e sovrascrivendolo se esiste già), dopodichè legge il file input.txt e ne trascrive su output.txt *i soli caratteri di posto pari*, ossia il secondo, il quarto, il sesto e così via. I caratteri di posto dispari vengono ignorati e sostituiti con delle “X”. Per esempio, se input.txt contiene la frase “prova testo da convertire”, al termine dell’esecuzione output.txt dovrà contenere “XrXvX XeXtX XaXcXnXeXtXrX”. Al termine devono essere chiusi entrambi i files. Se invece non era stato possibile aprire uno dei due files, il programma deve segnalarlo e terminare.

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

```
int main(int argc, char *argv[])
{
    FILE * infile;
    FILE * outfile;
    int car;
    infile=fopen("input.txt","r");
    if(infile==NULL){printf("no input file");return;}
    outfile=fopen("output.txt","w");
    if(outfile==NULL){printf("no output file");return;}
    while(!feof(infile))
    {
        if(fscanf(infile,"%c",&car)<0)break;
        fprintf(outfile,"X");
        if(fscanf(infile,"%c",&car)<0)break;
        fprintf(outfile,"%c",car);
    }
    fclose(infile); fclose(outfile);
    system("pause");
    return 0;
}
```

Politecnico di Milano - Anno Accademico 2005-06 - Informatica C
Appello 10 luglio 2006 – A

COGNOME e NOME	
Matricola	

Es 2. [punti 3]

Scrivere il main di un programma prova.exe che, se è stato specificato esattamente un parametro su riga di comando, lo analizza e comunica se la lunghezza è pari o dispari; inoltre, verifica se all'interno della stringa è presente la sequenza di caratteri "ab" (minuscoli e consecutivi senza spazi interposti), nel qual caso stampa un apposito messaggio. Se il numero di parametri è eccessivo o insufficiente, il programma lo segnala e termina. Il comportamento deve essere come quello qui accanto mostrato.

```
C:\>prova sabato
Lunghezza pari
Contiene la sequenza 'ab'
```

```
C:\>prova pippo
Lunghezza dispari
```

```
C:\>prova
Numero errato di parametri
```

```
int main(int argc, char *argv[])
{
    int i,cnt=0;
    if(argc!=2)
    {
        printf("Numero errato di parametri\n");
        return;
    }
    if(strlen(argv[1])%2)printf("Lunghezza pari\n");
    else printf("Lunghezza dispari\n");
    for(i=0; i<strlen(argv[1]); i++)
        if((argv[1][i]=='a') && (argv[1][i+1]=='b'))
            printf("Contiene la sequenza 'ab'\n");
    return 0;
}
```

Politecnico di Milano - Anno Accademico 2005-06 - Informatica C
Appello 10 luglio 2006 – A

COGNOME e NOME

Matricola

Es 3a. [punti 2]

Definire un tipo di dato idoneo a fungere da nodo di lista singola con capacità di contenere valori di tipo intero.

Definire inoltre una variabile **head**, di tipo idoneo a contenere un puntatore alla testa della lista.

```
struct nodo_s{
    int valore;
    struct nodo_s * next;
};

struct nodo_s * head;
```

Es 3b. [punti 5]

Scrivere una funzione `cerca`, che prende un parametro di tipo intero e ritorna un risultato di tipo intero, che verifica, all'interno della lista, se è presente un nodo caricato con il valore ricevuto come parametro. Se è presente in lista almeno un nodo che soddisfa tale condizione, la funzione ritorna 1. Altrimenti ritorna 0. Non si assuma che la lista contenga valori ordinati in modo crescente.

```
int cerca(int valore)
{
    struct nodo_s * scan;
    scan=head;
    while(scan)
    {
        if(scan->valore==valore) return 1;
        scan=scan->next;
    }
    return 0;
}
```

Politecnico di Milano - Anno Accademico 2005-06 - Informatica C
Appello 10 luglio 2006 – A

COGNOME e NOME	
Matricola	

Es 4. [totale max punti 3]

Ogni risposta esatta vale 1 punto. Ogni risposta errata vale -0.25 punti.
Ogni risposta non data vale 0 punti.
NOTA: Nel caso di quesiti a risposta chiusa, tra le risposte proposte per la domanda una sola e' esatta.

.....

4a

```
unsigned char vettore[10];  
vettore[10]=255;
```

- il frammento di programma sarà rifiutato dal compilatore
- il frammento di programma sarà accettato dal compilatore ma in esecuzione potrà causare errori
- il frammento di programma e' corretto

.....

4b

```
#include <stdio.h>  
int main(int argc, char *argv[])  
{  
    char buffer[100];  
    printf("%d", strcmp(buffer, "prova"));  
}
```

- il programma contiene un errore sintattico
- il programma è corretto ma è impossibile prevedere con certezza il risultato
- il programma è corretto e il risultato è 1

.....

4c

Descrivere il comportamento di questo frammento di programma.

```
int i;  
for(i=0; i<10; i++)  
{  
    printf("%d\n", i);  
    if(i=5) /* errore: occorre l'operatore == */  
    {  
        printf("siamo a metà ciclo\n");  
    }  
}
```

Il programma non termina mai, a causa dell'errore nell'espressione dell'if. Dopo una prima iterazione, sullo schermo apparirà infinite volte il messaggio "siamo a metà ciclo" insieme al valore 6.