SPIEGAZIONE DEL PROGRAMMA HUFFMAN

Il programma può essere suddiviso nelle tre macro aree sequenziali: analisi quantitativa del testo, estrazione dei periodi ed infine esecuzione dell'algoritmo di Huffman, esse sono applicate dalla funzione *main* che svolge il ruolo di controllore (assumiamo il suo punto di vista); descriviamo nel dettaglio le varie fasi.

Analisi qualitativa

Nella prima parte di questa fase si apre il file di testo tramite due puntatori distinti (*fd* e *fp)*, per poi passare alla fase di analisi che prevede la chiamata alla funzione *numeri_righe_frasi*: essa restituisce un array (monodimensionale) di due posizioni, per il numero di righe e per il numero di periodi del testo.

array_righe_frasi =

0	1
n. righe	n. frasi

E poi si determinano il numero di caratteri presenti in ogni riga ed in ogni periodo, escludendo il carattere terminatore di riga ed il carattere ".", tramite la chiamata alla funzione *numeri_caratteri* che restituisce l'array (bidimensionale) di un numero di celle per la riga 0 pari a n.righe ed un numero di celle per la riga 1 pari a n. frasi; in particolare si osservi che in corrispondenza dell'indice di colonna si farà riferimento alla riga 0 (o frase 0), riga 1 (o frase 1), ad esempio.

array_caratteri =

	0	1	2	3	4	…
	char riga 0	char riga 1	char riga 2	char riga 3	char riga 4	…
	char frase 0	char frase 1	char frase 2	char frase 3	char frase 4	…

Osserviamo che le due righe dell'array non è detto che siano della stessa lunghezza.

Infine si inizializza la matrice vuota bidimensionale, costruita ad hoc per contenere i periodi del testo, distribuiti sulle sue righe.

array_bid =

	0	1	2	3	4	…
frase 0	char 0 di frase 0	char 1 di frase 0	char 2 di frase 0	char 3 di frase 0	char 4 di frase 0	
frase 1	char 0 di frase 1	char 1 di frase 1	char 2 di frase 1	char 3 di frase 1	char 4 di frase 1	
⋮						

Osservazione: Nota importante sul programma

Non è chiaro ma per qualche motivo ignoto il programma sovrascrive i parametri determinati nella fase "analisi quantitativa", sporcandoli con valori casuali e senza un apparente significato. Ragion per cui, a seguito di numerosi tentativi di correzione, si è deciso di costellare il programma di ridondanze, atte a risettare i valori corretti, andando a risovrascrivere il contenuto delle variabili; le ridondanze sono racchiuse fra stringhe di commento per "isolarle" dal resto del programma.

Estrazione delle frasi

La fase è circoscritta ad un ciclo *while*, in cui si utilizza il secondo puntatore al file per eseguire l'estrazione di ogni singola riga del testo: ad ogni ciclo dell'operatore viene alloccata un array (monodimensionale), con la dimensione presa da *array_caratteri*, e gli si viene inserita la corrispondente riga del testo; successivamente l'array temporaneo viene inviato alla funzione *estrazione* che ne estrae le frasi.

Due puntatori indipendenti scorrono sulla riga, tenendo traccia della posizione sulla riga e della posizione sulla frase (nella riga attuale), con i quali si possono giungere ad una delle possibili situazioni seguenti:

- La/e frase/i iniziano e si concludono nella riga corrente, allora la funzione provvede a salvarle nell' *array_bid*; in corrispondenza delle loro posizioni prestabilite e si passa alla riga successiva, modificando i puntatori.
- La frase inizia nella riga attuale ma si conclude in una delle righe successive (non per forza quella immediatamente dopo), allora la funzione salva parte della frase nell' *array_bid*, conserva la posizione in cui è arrivato il puntatore nella frase e passa alla riga successiva; da essa si prosegue nella scansione della frase, con relativo salvataggio.

Questa fase si conclude con l'aver riempito l'*array_bid* con le frasi del testo, si osservi che nelle frasi non prefigura il carattere ".".

Esecuzione Huffman

Ultima fase del programma che si divide nelle due chiamate del *main* alle funzioni *frequenza* e *ordina_stampa*.

Frequenza

La funzione passa in argomento la singola lettera, di ogni frase, ad un thread che esegue la *thread_function*: la lettera, essendo una variabile di tipo *char* viene identificata sulla base della tavola del codice ASCII, in particolare per lettere maiuscole:

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V

87	88	89	90
W	X	Y	Z

Mentre per lettere minuscole:

97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r

115	116	117	118	119	120	121	122
s	t	u	v	w	x	y	z

Quindi, in base alla lettera verrà incrementato il valore della cella corrispondente, nell'array *contatore_caratteri*, che terrà conto della molteplicità; si osservi che tale array avrà la forma:

0	1	2	3	4	5	6	7	8	9	10	11	12
a	b	c	d	e	f	g	h	i	j	k	l	m
mol. di a	mol. di b	mol. di c	mol. di a	mol. di d	mol. di f	mol. di g	mol. di h	mol. di i	mol. di j	mol. di k	mol. di l	mol. di m

13	14	15	16	17	18	19	20	21	22	23	24	25
n	o	p	q	r	s	t	u	v	w	x	y	z
mol. di n	mol. di o	mol. di p	mol. di q	mol. di r	mol. di s	mol. di t	mol. di u	mol. di v	mol. di w	mol. di x	mol. di y	mol. di z

Ed infine ogni molteplicità di ogni lettera viene diviso per il numero totale di tutte le lettere del testo, questo per determinare la loro frequenza.

Ordina_stampa

La fase finale prevede solo di ordinare, in ordine decrescente, le frequenze determinate nella fase precedente, ed esso è stato realizzato implementando l'algoritmo di ordinamento "Insertion Sort", per poi stampare il risultato ottenuto.