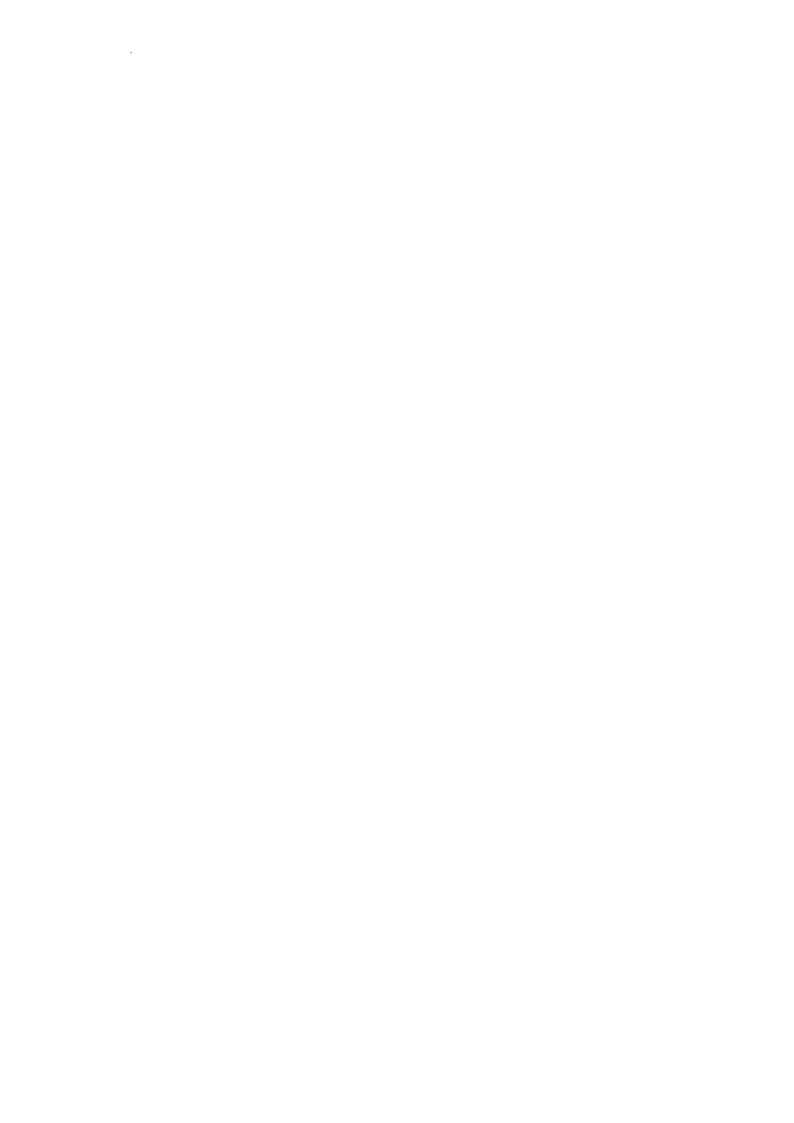
•	do-while	

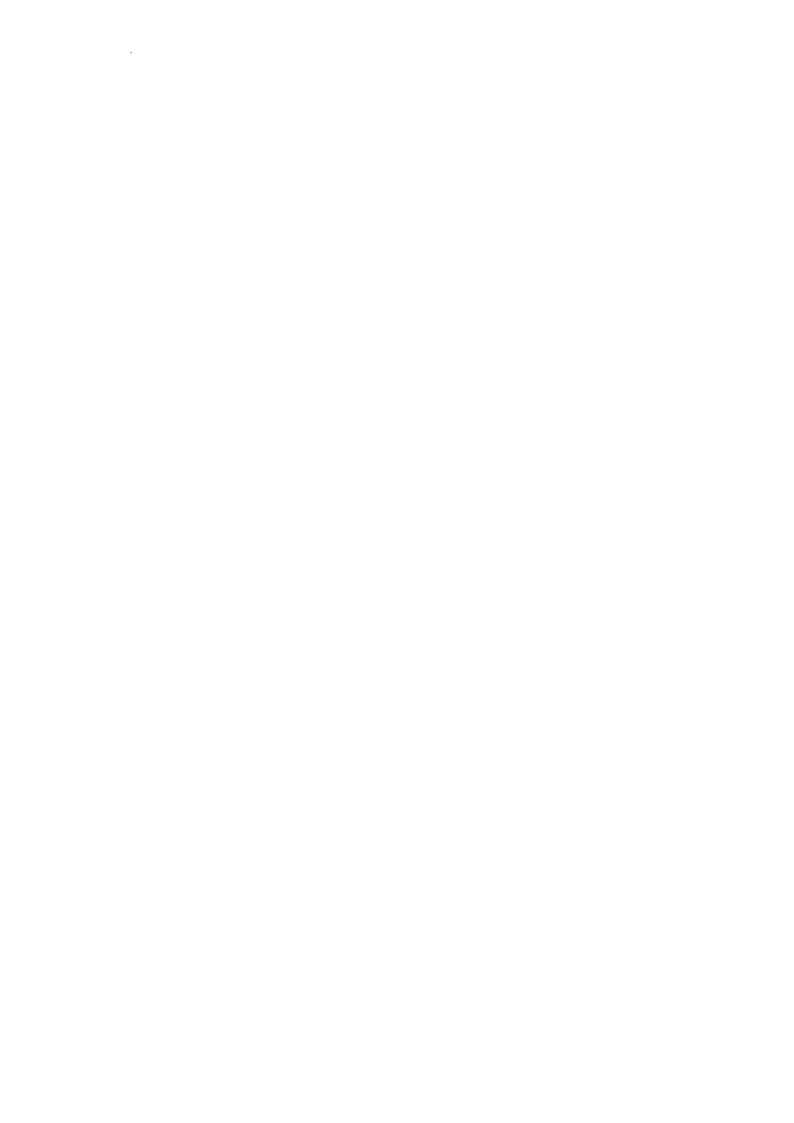
139
14

		·			 ·





```
i = 0;
while ((to[i] = from[i]) != '\0')
```



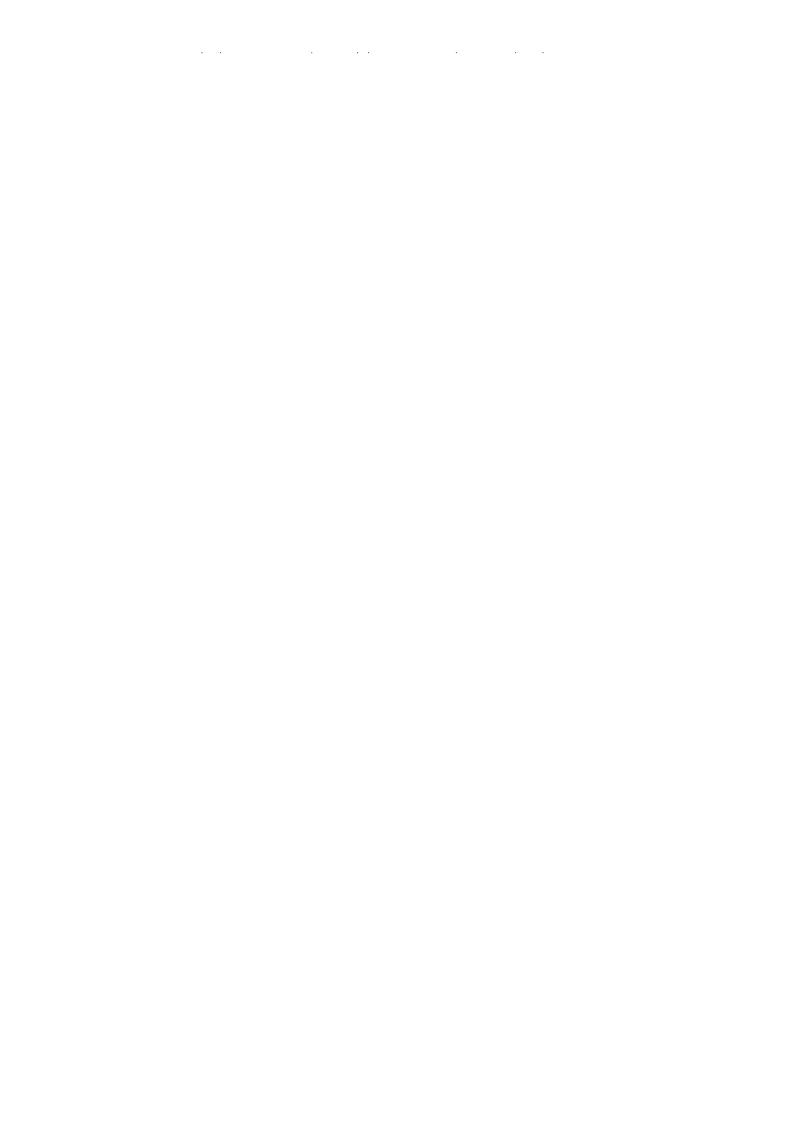
and the control of the

1

for (i = 0; i < lim-1 && (= getchar()) != EOF && != '\n'; ++i)

for (b = 0; x != 0; x >>= 1) if (x & 01) z = (a > b) ? : b; /* z = max(a, b) */



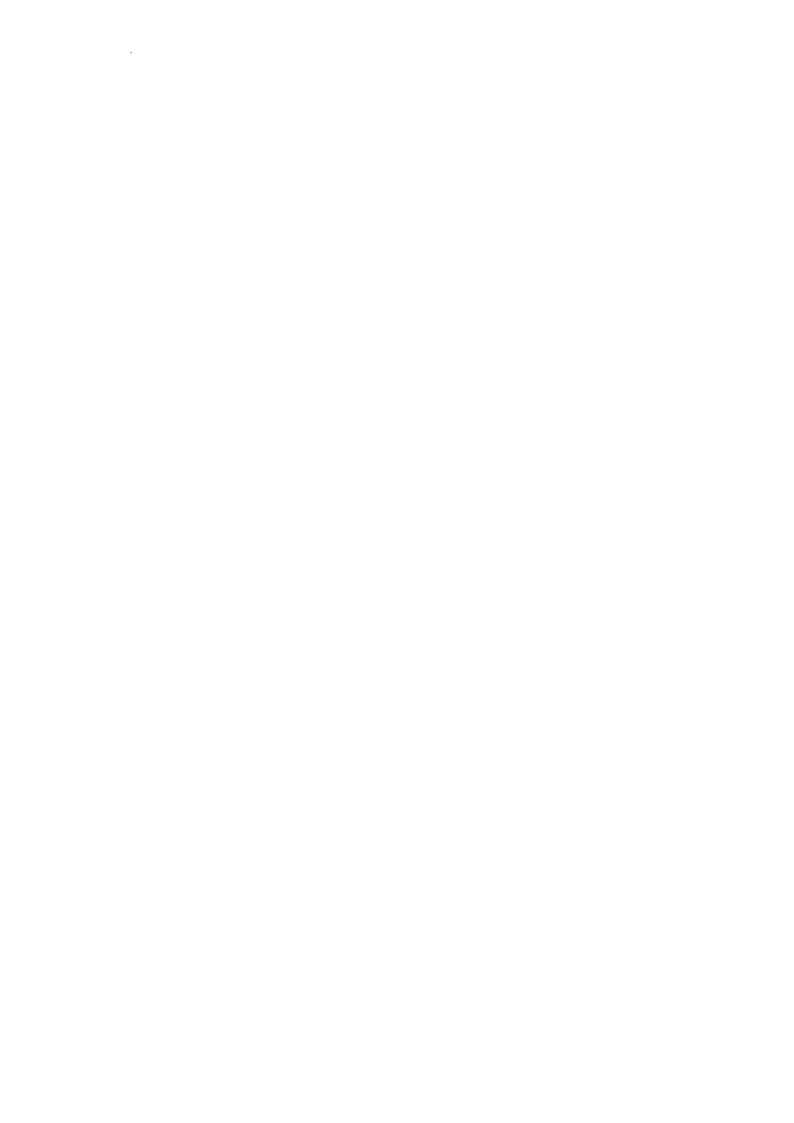




while

```
if (s[i] =='.')
    i++;
```


```
push (atof (s));
  break;
case '+':
  push (pop() + pop());
```



			·		



```
int n, array[SIZE], getint (int *);
for (n = 0; n < SIZE && getint (&array[n]) != EOF; n++)
;</pre>
```



*__

* ++ = val

nlines = 0;

```
{
    int i, leap;
    leap = year%4 == 0 && year%100 != 0 i I year%400 == 0;
```





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

```
void *temp;
temp = v[i];
v[i] = v[j];
v[j] = temp;
}
```

int *(

```
int tokentype; /*
char token[MAXTOKEN]; /*
char name[MAXTOKEN]; /*
char datatype[MAXTOKEN]; /*
char out[1000]; /*

*/
*/
*/
```

getch ungetch

rect point.

```
p1.y += p2.y;
return p1;
}
```

```
char *word;
int count;

struct key {
    char *word;
    int count;
} keytab[NKEYS];

    key keytab
```

```
;
if (c != EOF)
    *w++ = c;
if (!isalpha(c)) {
    *w = '\0';
    return c;
```

```
#define MAXWORD 100

struct tnode *addtree(struct tnode *, char *);
void treeprint(struct tnode *);
int getword(char *, int);

/*
main()
{
```

#include <</pre>

```
for (np = hashtab[hash(s)]; np != NULL; np = np->next)
   if (strcmp(s, np->name) == 0)
      return np; /* */
return NULL
```

typedef,

```
struct {
   char *name;
   int
```

 putchar(c)	· c	•	

:%-15.10s: :hello, wor :

printf

```
switch (*++ ) {
```

```
scanf("%d", &n);
scanf("%d", n);
```

		 	·		

return

lseek(fd, OL, O);

· OF

char *base

stbuf

```
}
closedir(dfd);
}
```

short	long		
•			



```
float y[4][3] = {
```

typedef- enum-







#

<stdlib.h

void *malloc(size_t size)
 malloc



SIGABRT

SIGFPE

SIGILL

INT_MIN

-32767