

Cs352 — Midterm

Date: Oct. 9, 2003

Time: 75 min

Name: _____ Login ID: _____

1. You should answer question 5, and EXACTLY 3 out of the questions 1,2,3,4. (if more questions are answered, the decision which ones would be checked is arbitrary).
2. Each question worth 25 points.
3. Write clearly. An answer that is not clearly written might not be checked.
4. The use of any books or notes is allowed.

1. (a) Write a macro named `c2v` that accepts a single character, assumed to be a digit, and returns the numeric value of this digit. For example, the command
`printf("%d" , c2v('2') + 1);`
should print **3**.
- (b) Use the macro `c2v` to write another macro `C2V` that accepts a character, and does the following. Once apply on a character which is not a digit, it should return the numeric value -1. Otherwise it should return the numeric value of the argument. For example `printf("%d %d" , c2v('2') + 1 , c2v('r') + 1);`
should print **3 0**.

Answer:

```
#define c2v(c) (c - '0')  
#define C2V(c) ((c >= '0' && c <= '9') ? (c2v(c)) : -1)
```

2. (a) Write a function called **SetBit**, whose prototype is `void SetBit(int *x, int k)`. The function should set the k 'th bit from the right of binary representation of the integer pointed by x to be 1 (the right-most bit is regarded as the 1st bit). So for example, **SetBit**(x ,1) should set the least significant bit of the integer pointed by x to 1. (its previous value is irrelevant).

Answer: $x = (01 \ll (k-1)) \mid x$

- (b) Write a function called **CheckBit**, whose prototype is `int CheckBit(int x, int k)`. The function should return 1 if the k 'th bit from the right of the binary representation of the variable x is 1. The function should return 0 otherwise.

Answer: $\text{return } (((01 \ll (k-1)) \& x) > 0)$

3. The following function opens the file “numbers.inp” for reading and modifying. It assumes that this file contains only integers, separated by whitespaces. The function reads these numbers, sums them, and writes the result of the summation at the end of the file. Finally the function needs to close the file.

Add the missing details in the places marked by underlines.

```
int x, sum =0;
_____fnp = fopen( “ _____” , “ _____” );
if (fnp == _____)
    { printf(“File cannot be opened” ) ;
      return 1 ;
    }
while ( fscanf( _____, “%d ”, _____) == _____)    sum += x ;
fprintf( _____, “ _____” , sum ) ;
_____ ( fnp ) ;
```

Answer:

Blank 1: FILE *

Blank 2: numbers.inp

Blank 3: r+

Blank 4: NULL

Blank 5: fnp

Blank 6: & x

Blank 7: 1

Blank 8: fnp

Blank 9: %d

Blank 10: fclose

4. (a) i. What is the output of the following program if the input from the user is **1234567890Q** ? (read the code carefully).

```
typedef struct cell {
    char c ;
    struct cell * next ;
} CELL ;

main(){
    CELL * p1 = NULL , *p2;
    char c ;

    while( (c=getchar() ) != 'Q' ) {
        p2 = (CELL *) malloc( sizeof(CELL)) ;
        p2-> next = p1 ;
        p2-> c = c ;
        p1 = p2 ;
    }

    for( p2 = p1 ; p2 != NULL && p2 -> next != NULL;
        p2 = p2->next ->next )
        printf(">> %c" , p2->c ) ;

    /* ADD CODE FOR NEXT QUESTION HERE */

}
```

- ii. Add to the existing code (at the place of the comments) that returns all allocated memory to the operating system.

(b) What is the output of the following program

```
char s[]="HelloWorld" ;  
char *p=3+s ;  
printf("%s", p ) ;
```

Answer:

(a.i) >> 0>> 8>> 6>> 4>> 2

(a.ii)

```
for (p2=p1; p2!=NULL; p1=p2) {  
    p2 = p2->next;  
    free(p1);  
}
```

(b) loWorld

5. Write a function called “middleOne” whose prototype is `int * middleOne(int *p1, int *p2)` . Once called, the pointer p1 points to the first element of an array of integers, and p2 points to the last element of this array. It is known that each element in the array is either 0 or 1, and that the number of 1’s is odd. The function should return a pointer to the element in the array that contains 1, and the number of 1’s to the left and to the right of this element are equal. Note that you do not know the number of 1’s. For full credit, your program must NOT use any variables except p1 and p2. You can assume that when `middleOne` is called, `*p1 = 0` and `*p2 = 0`.

Answer:

```
while (p1 < p2) {  
    .   do  p1++   while ( *p1! = 1 ) ;  
    .   do  p2--   while ( *p2! = 1 ) ;  
}  
return p1
```