



```

floatnew *new_floatnum(float f, int i) {
    _____; // L1: Allocate memory for
a floatnew in a temporary variable called temp
    _____; // L2: Initialise the i
field of the floatnew as per the function arguments
    _____; // L3: Initialise the f
field of the floatnew as per the function arguments
    return temp;
}

int main() {
    floatnew *f1 = new_floatnum(1.0, 2);
    // Do something with f1
    free(f1);
    return 0;
}

```

4. An array (a) contains pairwise positive and negative numbers with same magnitude in continuation. Among such n-pairs, only one pair does not match. Write a code to determine which pair of numbers are not matching in magnitude. E.g., if a[] = {10, -10, 20, -20, 30, -30, 40, -45}, mismatched numbers are 40 and -45. The code should work for any number of pairs in the array. **[2+2+2 = 6M]**

```

#define pair 5
int Matching(int *a, int n) {
    int i,j, flag=-1;
    for(i=0;i<n; i++)
    {
        if(_____)
        {
            _____
        }
    }
    return flag;
}

int main(){
    int a[]={10, -10, 20,-20, -30, 30, 40, -45, -50, 50}, k;
    k=Matching(a, pair);
    if(k>=0) printf("Mismatched elements:%d and %d\n",_____);
    else    printf("all pairs are matched\n");
    return 0;
}

```

5. The following incomplete program searches for a city name (entered by the user) in a text file "city.txt" and prints the position in the file where it is found. Fill in the blanks appropriately. **[8\*2 = 16M]**

```

int stringCompare(char *a, char *b) {
    int flag=0;
    while(*a!='\0' && _____) {
        if(_____)
            _____
        _____
    }
}

```

```

if(_____)
    return 0;
else
    return 1;
}

int main(){
    char buf[100], str[20];    int temp,    pos=-1, h, i=0;
    printf("\nEnter the string to be searched: ");
    scanf("%s",str);

    FILE * fp = fopen("city.txt", "r");
    while(fgets(_____) != NULL)
    {
        temp=stringCompare(buf, str); i++;
        if(temp==0)
        {
            _____
            _____
        }
    }
    if(pos != -1) printf("City name is found at %d position\n", pos);
    else printf("City name is not found in the file!\n");
    fclose(fp);
    return 0;
}

```

6. The output of the following piece of code is \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ when CHECK is 1, 2 and 3 respectively. Note that the CHECK macro can be changed to 1 or 2 or 3. [2+2+2 = 6M]

```

#define CHECK 1
int main() {
    int Data1 = 1896;
    int Data2 = 1977;
    #if CHECK==1
    int const *piData = &Data1;
    Data1++;
    #elif CHECK==2
    int const *piData = &Data1;
    *piData = 3;
    #elif CHECK==3
    int* const piData = &Data1;
    printf("*piData is %d\n",*piData);
    piData = &Data2;
    #endif
    printf("Data is %d",*piData);
}

```

7. The following C function takes a singly linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Fill in the blanks. [2+2+2 = 6M]

```

struct node {
    int data;
    struct node* next;
};
//head refers to the first node of the list
struct node *move_to_front(struct node *head) {
    struct node *p, *q;

```

```

if((head == NULL) || (head->next == NULL))
    return head;
q = NULL; p = head;
while (p->next != NULL) {
    q = p;
    p = p->next;
}

_____
_____
_____

return head;
}

```

(8-16) What will be the outputs of the following codes? Assume that the necessary standard libraries, and the wrapper (main) functions already exist. **[9\*2 = 18M]**

<pre> 8. int q(int a, int b) {     if (a == b)         return 1;     else if ( a % b )         return q(a, b+1);     else         return 0; } int main(void) {     printf("%d %d %d\n", q(35,2), q(11,2), q(23,2)); } _____ </pre>	<pre> 9. void find(int n) {     if (n == 0) return;     printf("%d", n%2);     find(n/2);     printf("%d", n%2); } void main(void) {     find(12); } _____ </pre>	<pre> 10. void fun(int (*b)[4]) {     b[1][1] = 100;     b++;     b[0][1] = 10; } void main() {     int a[2][4] = {{1,2,3,4}, {5,6,7,8}};     fun(a);     printf("%d %d", a[0][1], a[1][1]); } _____ </pre>
<pre> 11. int a[10], *p, *q; a[9]=11; p=&amp;a; q=a+9; printf("%d %d %d %d", a[9], *(p+9), *q, q-p); _____ </pre>	<pre> 12. char a[7]="PILANI"; int m=1; for( ; a[m]!='\0'; m++)     if((m%2)!=0)         printf("%c",a[++m]); _____ </pre>	<pre> 13. int i, j; for (i = 0; i &lt; 5; ++i) {     for (j = 0; j == i; ++j) {         printf("%d", j);     } } _____ </pre>
<pre> 14. #define fun1(x) 50 #define fun2(x) (fun1(x)&lt;0) ? fun1(x)-5 : fun1(x)+5 int main() {     int x=5;     printf("\nfun: %d",fun2(++x+3));     return 0; } _____ </pre>	<pre> 15. int i, j, k=12; for (i = 2; i &lt;= k; ++i) {     for (j = 2; j &lt;= i; ++j) {         if (i == j)             printf("%d", i);         else if (i%j == 0)             break;     } } _____ </pre>	<pre> 16. int a[][3] = {0, 1, 2, 3, 4, 5}; int (*ptr)[3] = a; ++ptr; printf("%d %d\n", (*ptr)[0], (*ptr)[1]); _____ </pre>

\*\*\*\*\*