Il programma ci permette, tramite un menù di scelta, di eseguire tre azioni differenti. Si può moltiplicare due numeri, dividerli o inserire una stringa.
Nel codice ci sono diversi errori.

```
char scelta = {'\0'};
menu ();
scanf ("%d", &scelta);
```

Nell'immagine vediamo dichiarata la variabile "scelta" come una variabile "char", ma poi nella funzione di inserimento viene usato un argomento errato. Viene usato "%d", che sta a indicare variabili di tipo intero, invece di "%c" che indica appunto una variabile di tipo carattere. Inoltre se viene inserito un input diverso dalle tre scelte date il programma termina, perché non sono state gestite casistiche diverse. Per risolvere si potrebbe usare un ciclo while con un ciclo if all'interno che  controllerà se l'input è corretto.

```
void moltiplica ()
{
        short int  a,b = 0;
        printf ("Inserisci i due numeri da moltiplicare:");
        scanf ("%f", &a);
        scanf ("%d", &b);

        short int prodotto = a * b;

        printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}
```

Nella funzione "moltiplica" c'è un errore di sintassi . In genere short e short int sono sinonimi, poiché indicano entrambi un numero intero di 16 bit, quindi int potrebbe essere omesso. In ogni caso nelle funzioni scanf c'è un errore di sintassi nell'argomento. Per risolvere bisogna assegnare a entrambe le funzioni "%hd", che è il formato da specificare per una variabile di tipo short int, oppure si può togliere lo short lasciando solo int e mettendo "%d" ad entrambi.

```
int divisione = a % b;
```

Nella funzione "divisione" c'è un errore logico nell'operatore "%" che restituisce solo il resto dell'operazione, mentre "/" restituisce il risultato della stessa.

```
char stringa[10];
printf ("Inserisci la stringa:");
scanf ("%s", &stringa);
```

Nell’immagine vediamo la funzione “stringa” con una vulnerabilità buffer overflow. La vulnerabilità consiste nella sovrascrittura delle variabili nel programma quando un input è sovradimensionato rispetto al volume concesso. Infatti la stringa viene dichiarata con uno spazio massimo di 10 caratteri, ma non c’è niente che impedisca di inserirne di più. Per risolvere si può usare una funzione “fgets” posta prima della funzione di input “scanf”. Questa funzione non prenderà in considerazione tutto ciò che sarà scritto oltre il limite prestabilito. Inoltre non è presente un’istruzione per leggere la stringa.