

- 1) The code will overflow because ints are only one byte so they can only hold 2^8 values (0-255) and $175 + 100 = 275 > 255$ to fix this you could make all the variables ints.
- 2) int numStat = 3 makes a variable numState that can be changed #define just makes const numState = 3 so numState is just replaced with 3 during and can't be redefined in the program
- 3) During compilation the assembler converts the preprocessed code into assembly for the specific processor
- 4) a) a = 12
b) b = 512
c) a = 107
d) b = 256
e) unknown index 5 out of bounds
- 5) #include <stdio.h>
- ```
int main() {
 int i = 97;
 while (i <= 122) {
 printf("%c\n", i);
 i++;
 }
 return (0);
}
```

### Problem 6)

```
typedef struct {
 int id;
 char brand[50];
 float miles[10];
} car;

int main() {

 struct car car1;

 car1.id = 1;
 car1.brand = 0;
 for (int i = 0; i <= 9; ++i){
 car1.miles[i] = 0.0;
 }
}
```

### Problem7)

```
typedef struct {
 int id;
 char brand[50];
 float miles[10];
} car;

int main() {

 struct car car1;

 car1.id = 1;
 car1.brand = 0;
 for (int i = 0; i <= 9; ++i){
 car1.miles[i] = 0.0;
 }
 printf("Enter the brand of the car:");
 scanf("%s",car1.brand);

}
```

**Problem 8)**

```
void calcmilescar(car car){
 for (int i = 0; i < 9; ++i){
 car.miles[i+1] = car.miles[i]*2;
 }
}
```

or

```
void calcmilesref(car*car){
 for (int i = 0; i < 9; ++i){
 car->miles[i+1] = car->miles[i]*2;
 }
}
```

**Problem 9)**

```
#include <stdio.h>
#include <string.h>
#define MAXLENGTH 100

int main(){
 int len;
 char str[MAXLENGTH];
 int mul;

 scanf("%s", str);
 len = strlen(str);
 scanf("%d", &mul);

 while(!(mul >= 1 && mul<=8)){
 printf("Enter a shift between 1 and 8");
 scanf("%d", &mul);
 }

 for (int i = 0; i < len; ++i) {
 str[i]= str[i]+mul;
 }

 printf("The result is:%s", str);
 return(0);
}
```