```
#include <stdio.h>
#include<conio.h>

// Function to check if a number is palindrome
int isPalindrome(int num) {
   int originalNum = num;
   int reversedNum = 0;

   while (num > 0) {
      int digit = num % 10;
      reversedNum = reversedNum * 10 + digit;
      num /= 10;
   }

   return (reversedNum == originalNum);
}

// Function to sort an array in ascending order using bubble sort
void bubbleSort(int arr[], int size) {
            int i,j;
   for(i=0;i<size-1;i++){
      for(j=0;j<size-i-1;j++){
         if (arr[j] > arr[j + 1]) {
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }
}
```

```
int main() {
    int numbers[25];
    int numCount = 0;
    int i;

    printf("Enter 25 numbers:\n");
    for(i=0;i<25;i++){
        scanf("%d", &numbers[i]);
        numCount++;
    }

    int choice;

    do{
        printf("\nMenu:\n");
        printf("1. Display odd numbers\n");
        printf("2. Display palindrome numbers\n");
        printf("3. Display 4-digit numbers\n");
        printf("4. Display numbers in ascending order\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Odd numbers:\n");
                for(i = 0; i < numCount; i++) {
                    if (numbers[i] % 2 != 0) {
                        printf("%d ", numbers[i]);
                    }
                }
```

```
    printf("\n");
    break;

case 2:
    printf("Palindrome numbers:\n");
                                    for(i = 0; i < numCount; i++) {
        if (isPalindrome(numbers[i])) {
            printf("%d ", numbers[i]);
        }
    }
    printf("\n");
    break;

case 3:
    printf("4-digit numbers:\n");
    for(i = 0; i < numCount; i++) {
        if (numbers[i] >= 1000 && numbers[i] <= 9999) {
            printf("%d ", numbers[i]);
        }
    }
    printf("\n");
    break;

case 4:
    printf("Numbers in ascending order:\n");
    bubbleSort(numbers, numCount);
    for (i = 0; i < numCount; i++) {
        printf("%d ", numbers[i]);
    }
    printf("\n");
    break;
```

```
            case 5:
                printf("Exiting program.\n");
                break;

            default:
                printf("Invalid choice. Please select a valid option.\n");
        }
    } while (choice != 5);
    getch();
    return 0;
}
```

```
Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 1
Odd numbers:
2003 1601 2001 545 7895 545 9347 43 23 345457 5467 67

Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 2
Palindrome numbers:
4 2002 545 545 646

Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 3
4-digit numbers:
2002 2003 1601 2001 7895 9347 6754 5467

Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 4
```

```
Enter 25 numbers:
16
04
2002
0510
2003
1601
2001
234
545
7895
234
545
646
732

9347
43
234
54
23
6754
24356
345457
5467
78
67
```

```
Enter 25 numbers:
16
04
2002
0510
2003
1601
2001
234
545
7895
234
545
646
732

9347
43
234
54
23
6754
24356
345457
5467
78
67

Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice:
```

```
Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 4
Numbers in ascending order:
4 16 23 43 54 67 78 234 234 234 510 545 545 646 732 1601 2001 2002 2003 5467 6754 7895 9347 24356 345457

Menu:
1. Display odd numbers
2. Display palindrome numbers
3. Display 4-digit numbers
4. Display numbers in ascending order
5. Exit
Enter your choice: 5
Exiting program.
```