

Overview-

This program manages a blood groups collection, and perform the following operations: create, delete, search, display, & update blood groups. Blood group package is also a feature of it. This system has been developed on a menu-driven approach which is supported by continuous- user interaction until the user exit.

Structure Definition

```
struct BloodGroup {  
    char bloodType[10];  
    int quantity;  
};
```

The **BloodGroup** structure stores information about each blood group, including the blood type as a string and the quantity as an integer.

Function Implementations

1. Add Blood Group

```
void addBloodGroup(struct BloodGroup *bloodGroups, int *count) {  
  
    if (*count >= MAX_BLOOD_GROUPS) {  
        printf("Cannot add more blood groups, limit reached\n");  
        return;  
    }  
    printf("Enter blood type: ");  
    scanf("%s", bloodGroups[*count].bloodType);  
    printf("Enter quantity: ");  
    scanf("%d", &bloodGroups[*count].quantity);  
    (*count)++;  
    printf("Blood group added successfully.\n");  
}
```

This function adds a new blood group to the array. It first checks if the maximum number of blood groups has been reached. If not, it prompts the user to enter the blood type and quantity, then increments the count.

2. Delete Blood Group

```

void deleteBloodGroup(struct BloodGroup *bloodGroups, int *count, const char
*bloodType){
    int index = searchBloodGroup(bloodGroups, *count, bloodType);
    if (index == -1){
        printf("Blood group not found.\n");
        return;
    }
    for (int i = index; i < *count - 1; i++) {
        bloodGroups[i] = bloodGroups[i + 1];
    }
    (*count)--;
    printf("Blood group deleted successfully.\n");
}

```

This function deletes a blood group identified by bloodType. It searches for the blood group, and if found, shifts the subsequent elements to fill the gap and decrements the count.

3. Search Blood Group

```

int searchBloodGroup(struct BloodGroup *bloodGroups, int count, const char
*bloodType){
    for (int i = 0; i < count; i++){
        if (strcmp(bloodGroups[i].bloodType, bloodType) == 0){
            return i;
        }
    }
    return -1;
}

```

This function searches for a blood group by its type. It returns the index if found, otherwise returns -1.

4. Create Blood Group Package

```

void createBloodGroupPackage(struct BloodGroup *bloodGroups, int count, const
char *packageDetails){
    printf("Creating blood group package with details: %s\n", packageDetails);
}

```

```
    displayBloodGroups(bloodGroups, count);  
}
```

This function is intended to create a package of blood groups, but it currently only prints the package details and displays the available blood groups.

5. Display Blood Groups

```
void displayBloodGroups(struct BloodGroup *bloodGroups, int count) {  
    if (count == 0) {  
        printf("No blood groups available.\n");  
        return;  
    }  
    printf("Blood Groups:\n");  
    for (int i = 0; i < count; i++) {  
        printf("Blood Type: %s, Quantity: %d\n", bloodGroups[i].bloodType,  
bloodGroups[i].quantity);  
    }  
}
```

This function displays all the blood groups and their quantities. If no blood groups are available, it prints an appropriate message.

6. Update Blood Quantity

```
void updateBloodQuantity(struct BloodGroup *bloodGroups, int count, const char  
*bloodType, int quantity) {  
    int index = searchBloodGroup(bloodGroups, count, bloodType);
```

```

    if (index == -1) {
        printf(" Blood group not found.\n");
        return;
    }
    bloodGroups[index].quantity = quantity;
    printf("Blood quantity updated successfully.\n");
}

```

This function updates the quantity of a specified blood group. It searches for the blood group and updates its quantity if found.

Main Function

```

int main() {
    struct BloodGroup bloodGroups[MAX_BLOOD_GROUPS];
    int count = 0;
    int choice;
    char bloodType[10];
    int quantity;
    while (1) {
        printf("\nBlood Bank Management System\n");
        printf("1. Add Blood Group\n");
        printf("2. Delete Blood Group\n");
        printf("3. Search Blood Group\n");
        printf("4. Display Blood Groups\n");
        printf("5. Update Blood Quantity\n");
        printf("6. Create Blood Group Package\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
    }
}

```

```
scanf("%d", &choice);
```

```
switch(choice){
```

```
case 1:
```

```
    addBloodGroup(bloodGroups, &count);
```

```
    break;
```

```
case 2:
```

```
    printf("Enter blood type to delete: ");
```

```
    scanf("%s", bloodType);
```

```
    deleteBloodGroup(bloodGroups, &count, bloodType);
```

```
    break;
```

```
case 3:
```

```
    printf("Enter blood type to search: ");
```

```
    scanf("%s", bloodType);
```

```
    if (searchBloodGroup(bloodGroups, count, bloodType) != -1) {
```

```
        printf("Blood group found.\n");
```

```
    } else {
```

```
        printf("Blood group not found.\n");
```

```
    }
```

```
    break;
```

```
case 4:
```

```
    displayBloodGroups(bloodGroups, count);
```

```
    break;
```

```
case 5:
```

```
    printf("Enter blood type to update: ");
```

```

scanf("%s", bloodType);
printf("Enter new quantity: ");
scanf("%d", &quantity);
updateBloodQuantity(bloodGroups, count, bloodType, quantity);
break;
case 6:
    createBloodGroupPackage(bloodGroups, count, "PackageDetails");
    break;
case 7:
    exit(0);
default:
    printf("Invalid choice!\n");
}
}
return 0;
}

```

How the System Works

1. **User Menu:** The program presents a menu to the user with options to add, delete, search, display, and update blood groups, create a blood group package, or exit the program.
2. **Choice Handling:** Based on the user's choice, the program calls the appropriate function.
3. **Loop:** The menu is presented in a loop, allowing the user to perform multiple operations until they choose to exit.

This system provides basic functionality for managing a blood bank's inventory, allowing for CRUD (Create, Read, Update, Delete) operations on blood group data.

Output:

```
D:\CSE\Problem\Function\Assignment.exe

Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice:
```

```
"C:\Users\TAMJID-PC\Desktop\ASSIGNMENT CSE109\Blood Bank Management System.c.exe"

Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 1
Enter blood type: A+
Enter quantity: 23
Blood group added successfully.

Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 1
Enter blood type: A-
Enter quantity: 22
Blood group added successfully.

Blood Bank Management System
```

"C:\Users\TAMJID-PC\Desktop\ASSIGNMENT CSE109\Blood Bank Management System.c.exe"

```
7. Exit
Enter your choice: 2
Enter blood type to delete: A-
Blood group deleted successfully.
```

```
Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 3
Enter blood type to search: A+
Blood group found.
```

```
Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 3
Enter blood type to search: A-
Blood group not found.
```

"C:\Users\TAMJID-PC\Desktop\ASSIGNMENT CSE109\Blood Bank Management System.c.exe"

```
Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 4
Blood Groups:
Blood Type: A+, Quantity: 23
```

```
Blood Bank Management System
1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit
Enter your choice: 5
Enter blood type to update: A+
Enter new quantity: 50
Blood quantity updated successfully.
```


"C:\Users\TAMJID-PC\Desktop\ASSIGNMENT CSE109\Blood Bank Management System.c.exe"

Blood quantity updated successfully.

Blood Bank Management System

1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit

Enter your choice: 6

Creating blood group package with details: PackageDetails

Blood Groups:

Blood Type: A+, Quantity: 50

Blood Bank Management System

1. Add Blood Group
2. Delete Blood Group
3. Search Blood Group
4. Display Blood Groups
5. Update Blood Quantity
6. Create Blood Group Package
7. Exit

Enter your choice: 7

Process returned 0 (0x0) execution time : 210.478 s

Press any key to continue.