DÜZCE ÜNİVERSİTESİ

CE102-Algorithms And Programming II

PROJECT REQUIREMENTS (Sem. II, 2022/2023)

Project Title: Food Donation Management System

-Group 1-

Group Members

Emine Hatun Çakmak	211015041
Büşra Basan	211015003
Zeynep Can	211015060
Göksu Kesen	211015008
Kadir Ocak	211015017

INSTRUCTOR: PROFESSOR Dr. Imad Fakhri Taha Alyaseen

THE AIM OF THE PROJECT

"Our goal in the food donation system is to create a system where donors can donate either food or money to our system.

In the system, donors are asked for some information, such as name, surname, phone number, email, address, fax, etc. They are also asked whether their donation is in the form of money or food. If it is money, they need to provide the amount, whether it is a one-time donation or a recurring donation, the date of the donation, the country where the donation is made, the currency, and payment information. If it is food, they need to provide the type of product, its name, expiration date, delivery date, and the country where the donation is made.

Through the food donation system, donors can easily make their donations to our system, saving time and ensuring that their donations reach those in need in a reliable manner."

GENERAL WORK SHARING

Emine Hatun ÇAKMAK
is responsible for donor (personal donor and corporate donor).

Göksu KESEN
is responsible for money donation (depending on the once or monthly) .

Zeynep CAN
is responsible for food donation.

Büşra BASAN
is responsible for displaying donation depending money or food.

Kadir OCAK
is responsible for payment when the user donates the money.

"Everyone helped each other when we need"

Softcopy Code

```
#include <iostream>

#include <sstream>

#include <string>

#include <ctime>   // This is a library for accessing date and time

#include <cstdlib>

/*using the system function from the cstdlib library enables you to execute

operating system-specific commands in a way that promotes portability.

This means your code can run on different operating systems without requiring changes.*/


struct PersonalDonor // We use this struct to store personal information

{

        std::string Name; // Personal donor's first name

        std::string lastName; // Personal donor's last name

        std::string email; // Personal donor's email address

        std::string phone; // Personal donor's phone number

};


struct CorporateDonor // We use this struct to store corporate information

{

        std::string name; // Corporate donor's name

        std::string email; // Corporate donor's email address

        std::string phone; // Corporate donor's phone number

        std::string address; // Corporate donor's address

        std::string fax; // Corporate donor's fax number

};


// It is a struct that includes year, month and day about expire date and delivery date

struct Date {
    int expireDay;
    int expireMonth;
    int expireYear;
    int delivery_year;
    int delivery_month;
    int delivery_day;

};

//In this section we define food related information.
struct Food
{
    std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" };
    std::string selectedCountry;
    std::string foodType;
    std::string foodName;
    std::string amountFood;
    Date expireDate; // a struct inside a struct
    Date delivery{}; // a struct inside a struct
};



// This structure holds the necessary information for one-time money donation transaction.
struct MoneyOnce
{
    std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" }; // The string "donationCountry" contains the names of the countries to which donations can be made.
    std::string selectedCountry;                        // The variable "selectedCountry" holds the country selected by the user.
    int quantityOfTheDonation{};                        // The variable "quantityOfTheDonation" holds the donation amount.
```

```
        std::string donationCurrency[6]{ "EUR","USD","GBP","TRY","JPY","RUB" };          // The string "donationCurrency" contains the currencies users can choose for donations.

        std::string selectedCurrency;                                        // The variable "selectedCurrency" holds the currency selected by the user.


};



// This struct holds the necessary information for a recurring fundraiser once a month.

struct MoneyMonthly


{

        std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" }; // The string "donationCountry" contains the names of the countries to which donations can be made.

        std::string selectedCountry;                                         // The variable "selectedCountry" holds the country selected by the user.

        int  quantityOfTheDonation{};                                        // The variable "quantityOfTheDonation" holds the donation amount.

        std::string donationCurrency[6]{ "EUR","USD","GBP","TRY","JPY","RUB" };          // The string "donationCurrency" contains the currencies the user can choose for donation.

        std::string selectedCurrency;                                        // The variable "selectedCurrency" holds the currency selected by the user.

        int monthDay{};                                                      // The variable "monthDay" is used to select which day of each month the donation will be made.

};


// Function to clear the console screen

void clearConsole()

{

        std::cout << "\nPress enter to continue..."; // Prompting user to press enter

        std::cin.ignore(); // Ignoring any characters in the input buffer

        std::cin.get(); // Waiting for the user to press enter

        system("cls"); // Clearing the console screen

}
```

```
void addPersonalDonor(PersonalDonor& donor) // Function to take user input for personal donor
{
    std::cout << "Please enter your name: ";
    std::getline(std::cin, donor.Name); // Get the personal donor's first name
    std::cout << "Please enter your last name: ";
    std::getline(std::cin, donor.lastName); // Get the personal donor's last name
    std::cout << "Please enter your email address: ";
    std::getline(std::cin, donor.email); // Get the personal donor's email address
    std::cout << "Please enter your phone number: ";
    std::getline(std::cin, donor.phone); // Get the personal donor's phone number
}

void addCorporateDonor(CorporateDonor& donor) // Function to take user input for corporate donor
{
    std::cout << "Please enter your corporation's name: ";
    std::getline(std::cin, donor.name); // Get the corporate donor's name
    std::cout << "Please enter your email address: ";
    std::getline(std::cin, donor.email); // Get the corporate donor's email address
    std::cout << "Please enter your phone number: ";
    std::getline(std::cin, donor.phone); // Get the corporate donor's phone number
    std::cout << "Please enter your corporation's address: ";
    std::getline(std::cin, donor.address); // Get the corporate donor's address
    std::cout << "Please enter your corporation's fax number: ";
    std::getline(std::cin, donor.fax); // Get the corporate donor's fax number
}

// This function adds a money donation transaction based on the "MoneyOnce" struct.
// The function retrieves the donor country, donation amount and currency selected by the user.
```

```
void addMoney(MoneyOnce& DonationOnce)  // function overloading

{

againCountry1:

        std::cout << "Please enter the country which you donate to: " << "\n";     //We have enabled
the user to select the countries by using
        std::cout                                                   //the array defined in the structure section.*/
                << "Please do not use non - digit characters.It can cause some problems.\n"
                << "Press 1 for Germany\n"
                << "     2 for USA\n"
                << "     3 for England\n"
                << "     4 for Turkey\n"
                << "     5 for Japan\n"
                << "     6 for Russia\n";

        int choice;
        std::cout << "Selection: ";
        std::cin >> choice;
        std::cout << "\n";
        std::cout << "------------------------------------------------------------\n";

        //Here we have defined an array containing 6 countries. If the data entered by the user is
one of these 6 countries, the next question is passed.
        //If the entered data is from other than these 6 countries, the user is asked to enter the input
again.
        if (choice >= 1 && choice <= 6)
        {
                DonationOnce.selectedCountry = DonationOnce.donationCountry[choice - 1];
                // The variable "DonationOnce.selectedCountry" holds the country selected by the
user.
```

```
        }



        else
        {

                std::cout << "Wrong selection. Please try again.\n";
                goto againCountry1;
        }
againCurrency1:
        std::cout << "Please choose your country currency:\n"
                << "Please do not use non - digit characters.It can cause some problems.\n"
                << "Press 1 for EUR\n"
                << "      2 for USD\n"
                << "      3 for GBP\n"
                << "      4 for TRY\n"
                << "      5 for JPY\n"
                << "      6 for RUB\n";
        int choice2;
        std::cout << "Selection: ";
        std::cin >> choice2;
        std::cout << "--------------------------------------------------------\n";
        std::cout << "\n";


        //Here we have defined an array containing 6 currencies. If the data entered by the user is one of these 6 currencies, the next question is passed.
        //If the entered data is other than these 6 currencies, the user is asked to enter the input again.
        if (choice2 >= 1 && choice2 <= 6) {

                DonationOnce.selectedCurrency = DonationOnce.donationCurrency[choice2 - 1];
            // The variable "DonationOnce.selectedCurrency" holds the currency of the user's choice.
        }



        else {
                std::cout << "Wrong selection. Please try again.\n";
                goto againCurrency1;
                //The "goto" statement allows the user to return to the same menu when a wrong choice is made, until they make the right choice.
        }


        std::cout << "Please enter the amount of the money will you donate: ";
        std::cin >> DonationOnce.quantityOfTheDonation;
        // The variable "DonationOnce.quantityOfTheDonation" holds the donation amount entered by the user.


        std::cout << "\nYou are directed to the payment page.";
        //This function directs the user to the payment page.
}


// This function adds a money donation transaction based on the "MoneyMonthly" struct.
// The function retrieves the donor country, donation amount and currency selected by the user.
void addMoney(MoneyMonthly& DonationMonthly) // function overloading


{


again3:


        std::cout << "Please enter the country which you donate to: " << "\n"
                << "Please do not use non - digit characters.It can cause some problems.\n"
                << "Press 1 for Germany\n"
                << "     2 for USA\n"
                << "     3 for England\n"
                << "     4 for Turkey\n"
                << "     5 for Japan\n"
                << "     6 for Russia\n";
        int choice3;
        std::cout << "Selection: ";
        std::cin >> choice3;
        std::cout << "\n";
        std::cout << "----------------------------------------------------------\n";

        //Here we have defined an array containing 6 countries. If the data entered by the user is one of these 6 countries, the next question is passed.
        //If the entered data is from other than these 6 countries, the user is asked to enter the input again.
        if (choice3 >= 1 && choice3 <= 6)
        {

                DonationMonthly.selectedCountry = DonationMonthly.donationCountry[choice3 - 1];
                // The variable "DonationMonthly.selectedCountry" holds the country selected by the user.
        }

        else
        {

                std::cout << "Wrong selection. Please try again.\n";
                goto again3;

        }

again4:
        std::cout << "Please enter your country currency: " << "\n"
                << "Please do not use non - digit characters.It can cause some problems.\n"
                << "Press 1 for EUR\n"
                << "      2 for USD\n"
                << "      3 for GBP\n"
                << "      4 for TRY\n"
                << "      5 for JPY\n"
                << "      6 for RUB\n";
        int choice4;
        std::cout << "Selection: ";
        std::cin >> choice4;
        std::cout << "------------------------------------------------------------\n";
        std::cout << "\n";


        //Here we have defined an array containing 6 currencies. If the data entered by the user is
one of these 6 currencies, the next question is passed.
        //If the entered data is other than these 6 currencies, the user is asked to enter the input
again.


        if (choice4 >= 1 && choice4 <= 6)
        {


                DonationMonthly.selectedCurrency = DonationMonthly.donationCurrency[choice4 -
1];
                // The variable "DonationMonthly.selectedCurrency" holds the currency of the user's
choice.
        }


        else
        {
                std::cout << "Wrong selection. Please try again.\n";
                goto again4;
        }

        std::cout << "Please enter the amount of the money will you donate monthly:\n"
                << "Please do not use non-digit characters.It can cause some problems.\n";
        std::cout << "Money quantity: ";
        std::cin >> DonationMonthly.quantityOfTheDonation;
        std::cout << "\n";
        // The variable "DonationMonthly.quantityOfTheDonation" holds the donation amount entered by the user.
        std::cout << "Please enter which day of each month you would like to donate:" << "\n";
        std::cout << "Please enter a number between 1 and 28.\n(Otherwise that can cause some problems):\n";
        std::cout << "The day: ";
againDate:
        std::cin >> DonationMonthly.monthDay;
        // The variable "DonationMonthly.monthDay" holds the user on which day of each month to donate.
        if (DonationMonthly.monthDay >= 1 && DonationMonthly.monthDay <= 28) {
                std::cout << "The day you chose has been accepted.\n";

        }
        else
        {
                std::cout << "----------------------------------------------------------\n";
                std::cout << "The day you entered could not be accepted. " << "\n\n"
                    << "Please enter a number between 1 - 28: ";
                goto againDate;
        }
        std::cout << "\nYou are directed to the payment page.";
}
void addFood(Food& food); // We defined our function above for overloading.
void addFood(std::string& select) { //In this section, the function that is called when it wants to add a food type other than the choices presented below has been created.Function overloading.


        std::cout << "Please enter the type of food item: ";

        std::cin.ignore();// This code clears the previously entered entry.

        std::getline(std::cin, select);




}


void addTypeofthefooditems(std::string& selection) { //In this function, we printed the food types on the screen and set up the if-else loop to be used when choosing the food types.

        // By using Again, we made sure that the same question is asked again when he makes the wrong choice.

again:

        std::cout <<

                "Please select the type of food item (Enter capital letters between \"A\"and \"M\"): \n"

                "A)Meat, Chicken, Fish etc. \n"

                "B)Milk and Dairy Products\n"

                "C)Egg\n"

                "D)Legumes \n"

                "E)Canned Foods\n"

                "F)Frozen Foods \n"

                "G)Bakery Products\n"

                "H)Bread and Cereals \n"

                "I)Dried Foods \n"

                "J)Vegetables and Fruits\n"

                "K)Oil\n"

                "L)Water \n"

                "M)Other Food Types \n";
```

```
std::cout << "Selection: ";
std::cin >> selection;
if (selection == "A") {
        selection = "Meat, Chicken, Fish etc.";
        std::cout << "---------------------------------------------------------------------\n";
}

else if (selection == "B") {
        selection = "Milk and Dairy Products";
        std::cout << "---------------------------------------------------------------------\n";
}
else if (selection == "C") {
        selection = "Egg";
        std::cout << "---------------------------------------------------------------------\n";
}
else if (selection == "D") {
        selection = "Legumes";
        std::cout << "---------------------------------------------------------------------\n";
}

else if (selection == "E") {
        selection = "Canned Foods";
        std::cout << "---------------------------------------------------------------------\n";
}

else if (selection == "F") {
        selection = "Frozen Foods";
        std::cout << "---------------------------------------------------------------------\n";
}
else if (selection == "G") {
        selection = "Bakery Products";
        std::cout << "---------------------------------------------------------\n";
}
else if (selection == "H") {
        selection = "Bread and Cereals";
        std::cout << "---------------------------------------------------------\n";
}
else if (selection == "I") {
        selection = "Dried Foods";
        std::cout << "---------------------------------------------------------\n";
}
else if (selection == "J") {
        selection = "Vegetables and Fruits";
        std::cout << "---------------------------------------------------------\n";
}
else if (selection == "K") {
        selection = "Oil";
        std::cout << "---------------------------------------------------------\n";
}

else if (selection == "L") {
        selection = "Water";
        std::cout << "---------------------------------------------------------\n";
}
else if (selection == "M") {
        addFood(selection);
        std::cout << "---------------------------------------------------------\n";
}
else {
        std::cout << "Wrong selection. Please try again.\n";
        std::cout << "---------------------------------------------------------\n";
        goto again;
        }


}


void addFood(Food& food) { /*In this function, we called the function that determines the type of
food and we created this function to learn the name of the food,

        from which country the food will be sent, how much of the food will be donated, the expiry
date of the food and the delivery date.*/

again5:



        std::cout << "Please enter the country which you donate to: " << "\n";      //We have enabled
the user to select the countries by using the array defined in the structure section.

        std::cout << "ATTENTION! If you log in other than the number, you will have to log in to the
system from the beginning." << "\n"

                << "Press 1 for Germany\n"

                << "      2 for USA\n"

                << "      3 for England\n"

                << "      4 for Turkey\n"

                << "      5 for Japan\n"

                << "      6 for Russia\n";


        int choice;

        std::cout << "Selection: ";

        std::cin >> choice;

        std::cout << "\n";

        std::cout << "----------------------------------------------------------\n";

        if (choice >= 1 && choice <= 6)

        {

                food.selectedCountry = food.donationCountry[choice - 1];

        }


        else
        {
                std::cout << "Wrong selection. Please try again.\n";          //Again5 is used to run
the loop from the beginning when the wrong selection is made.
                goto again5;
        }


        addTypeofthefooditems(food.foodType);           //The function(Type of the food items)
was called.



        std::cout << "Please enter the name of the food item: ";    //The user is prompted to enter
the name of the food.
        std::cin.ignore(); // This code clears the previously entered entry.
        std::getline(std::cin, food.foodName);
        std::cout << "------------------------------------------------------------\n";
        std::cout << "Please enter the amount of donated food in kg/L/packet/number.\n"
                << "(Please enter just number): "; //The user is asked to enter how much the food
will donate.
        // This code clears the previously entered entry.
        std::getline(std::cin, food.amountFood);
        std::cout << "------------------------------------------------------------\n";
invalid:
        std::cout << "Please enter as number  the year of expiration date (enter as number): ";
//The user is asked to enter the expiry date of the food (year, month and day are asked as separate
questions).
        std::cin >> food.expireDate.expireYear;                              // In this section, it is
accessed the member of the struct inside the struct by using . (dot)
        std::cout << "Please enter as number the month of expiration date (enter as number
between 1-12): ";
        std::cin >> food.expireDate.expireMonth;
        std::cout << "Please enter as number the day of expiration date (enter as number between 1-
30): ";
        std::cin >> food.expireDate.expireDay;
```

```
	std::time_t t = time(nullptr);	//These codes are written to access the computer's date information.
	std::tm today;
	localtime_s(&today, &t);
	int todayDay = today.tm_mday;		//These codes are written to define the day, month and year correctly.
	int todayMonth = today.tm_mon + 1;
	int todayYear = today.tm_year + 1900;

	//If the expiration date entered in the if-else loop has passed (before today's date), the food name, how much the food will be donated, and the expiration date are requested.

	if (food.expireDate.expireYear < todayYear || (food.expireDate.expireYear == todayYear && food.expireDate.expireMonth < todayMonth) || (food.expireDate.expireYear == todayYear && food.expireDate.expireMonth == todayMonth && food.expireDate.expireDay < todayDay)) {

		std::cout << "You can't donate food because the expiration date has passed.\n" << std::endl;
		std::cout << "-----------------------------------------------------------\n";
		goto invalid;
	}
	//If the correct expiration date is written, the delivery date is asked (year, month and day are asked in separate questions).
	else {
		std::cout << "You can donate food that has not passed its expiration date." << std::endl;

	invalid2:
		std::cout << "-----------------------------------------------------------\n";
		std::cout << "Please enter as number the year of deliver date (enter as number ): "; // In this section, it is accessed the member of the struct inside the struct by using . (dot)
		std::cin >> food.delivery.delivery_year;
		std::cout << "Please enter as number the month of deliver date (enter as number between 1-12): ";
		std::cin >> food.delivery.delivery_month;
		std::cout << "Please enter as number the day of delivery date (enter as number between 1-30): ";
		std::cin >> food.delivery.delivery_day;



		//Checks whether the delivery date is before or after the expiration date. After the expiration date, it asks for a re-delivery date.
		if (food.delivery.delivery_year > food.expireDate.expireYear || (food.delivery.delivery_year == food.expireDate.expireYear && food.delivery.delivery_month > food.expireDate.expireMonth) || (food.delivery.delivery_year == food.expireDate.expireYear && food.delivery.delivery_month == food.expireDate.expireMonth && food.delivery.delivery_day > food.expireDate.expireDay)) {

			std::cout << "You can't deliver the food on this date because the delivery date is after the expiry date.\n" << "Please enter a new delivery date." << std::endl;

			goto invalid2;
		}

		else {
			//Checks whether the delivery date is before or after today's date. If it is before today's date, it asks for a re-delivery date.

			while (food.delivery.delivery_year < todayYear || (food.delivery.delivery_year == todayYear && food.delivery.delivery_month < todayMonth) || (food.delivery.delivery_year == todayYear && food.delivery.delivery_month == todayMonth && food.delivery.delivery_day < todayDay)) {
				std::cout << "You entered a past date.\n" << "Please enter a new delivery date." << std::endl;
				goto invalid2;
			}
			std::cout << "------------------------------------------------------------\n";
			std::cout << "You can deliver the food on" << " " << food.delivery.delivery_year << "/" << food.delivery.delivery_month << "/" << food.delivery.delivery_day << std::endl;
		}
	}
}

// This a function that displays on the screen the amount of the food, the name of the food and country where is sent that have donated
void displayDonationFood(Food* ptr2) // The parameter of the function is a struct
{
	std::cout << "You have donated " << ptr2->amountFood << " kg/L/packet/number of " << ptr2->foodName << " from " << ptr2->selectedCountry << "." << std::endl;
	// I access the members of the Food struct by using pointer
	std::cout << "Thanks for your Donation...\n";
}

//This a a function that ddisplays on the screen  and the country where is sent that has donated once.
void displayDonationMoney(MoneyOnce* ptr2) // Function overloading
{
	std::cout << "You have donated " << ptr2->quantityOfTheDonation << ptr2->selectedCurrency << " amount of money once from " << ptr2->selectedCountry << "." << std::endl;
	// I access the members of the MoneyOnce struct by using pointer
	std::cout << "Thanks for your Donation...\n";
}

// This is a function that displays on the screen the quantity of the money and the country where is sent that has donated monthly and also it shows the day of the every month
void displayDonationMoney(MoneyMonthly* ptr2)  // Function overloading
{
	std::cout << "You will donate " << ptr2->quantityOfTheDonation << ptr2->selectedCurrency << " on " << ptr2->monthDay << " of the every month from " << ptr2->selectedCountry << "." << std::endl;
	// I access the members of the MoneyMonthly  struct by using pointer
	std::cout << "Thanks for your Donation...\n";
}
```

```
bool cardNumber(int number) //**Kadir -> Function to check if the card number is valid.
{
    if (number >= 1000 && number <= 9999) {
        return true;
    }
    return false;
}

void cardDonation(int cardRow, int cardColumns) // Function to process card donation.
{
    std::string expirationDate;
    std::string cvvCode;
    int cardNumbers[1][4];

    std::cout << "Please enter your card numbers ([4]-[4]-[4]-[4]): "; // Prompt the user to enter
card numbers.
    for (int a = 0; a < cardRow; a++) {
        for (int b = 0; b < cardColumns; b++) {
            std::cin >> cardNumbers[a][b];
            if (!cardNumber(cardNumbers[a][b])) // Check if the entered card number is
valid.
            {
                std::cout << "* Please re-enter your card number.\n"
                    << "* Make sure you have entered your card information
correctly. "
                    << std::endl;
                std::cout << "Please enter your card numbers: ";
                b--; // Decrease the value of 'b' to re-enter the invalid number.
            }
        }
    }
    std::cout << "Please enter expiration date of your card (MM-YY): ";  // Prompt the user to enter expiration date and CVV code.
    std::cin >> expirationDate;
    std::cout << "Please enter your CVV code (xxx): ";
    std::cin >> cvvCode;

    std::cout << std::endl << "Card numbers you have entered: ";
    for (int c = 0; c < cardRow; c++)  // Display the entered card numbers.
    {
        for (int d = 0; d < cardColumns; d++) {
            std::cout << cardNumbers[c][d] << " ";
        }
    }
    std::cout << std::endl << "Expiration Date:  " << expirationDate << std::endl;
    std::cout << "CVV code: " << cvvCode << std::endl;
    std::cout << "Thanks for your donation." << std::endl;

}//Payment Function

int main()
{

    std::string donorType;
againDonor:
    std::cout << "If you are a personal donor please press 1 or  a corporate donor press 2:";
    std::getline(std::cin, donorType); //We used getline so that it does not give an error if more than one character is entered in the user input.

    /* We use if block because of select one of the user type.
    If the user select a different type of character then the else function returned back to question line and repeat .*/
    if (donorType == "1")
    {
        PersonalDonor donor; // Create a personal donor object
        addPersonalDonor(donor); // Call the function to input personal donor information
        clearConsole();
    }
    else if (donorType == "2")
    {
        CorporateDonor donor; // Create a corporate donor object
        addCorporateDonor(donor); // Call the function to input corporate donor
information
        clearConsole();
    }
    else
    {
        std::cout << "You entered an invalid donor type. Please choose one of the provided
options.\n";
        goto againDonor; // If an invalid donor type is entered, repeat the question
    }

    const int cardRow = 1;
    const int cardColumns = 4;

    std::string donationTime;
    std::string donationType;
againDonation:

    std::cout << "Will you donate food or money?\nIf you donate food press 1 or money press 2 :
";
    std::cin >> donationType;
    std::cout << "----------------------------------------------------------\n";
    std::cout << "\n";
```

```
	if (donationType == "1")
	{
		Food food; //Here it passes to the food function.
		Food selection;
		addFood(food);
		clearConsole();
		displayDonationFood(&food); // Here,the function that displays food is invoked  by passing address of the Food Struct.
	}
	else if (donationType == "2")
	{

		while (true)
		{
			std::string donationTime;
			std::cout << "Will you donate money once or monthly?\nfor once press 1; for monthly press 2: ";
			std::cin >> donationTime;
			std::cout << "------------------------------------------------------------\n";
			std::cout << "\n";

			if (donationTime == "1")
			{
				MoneyOnce MoneyOnce;
				addMoney(MoneyOnce);
				clearConsole();
				cardDonation(cardRow, cardColumns); // Call the card donation function.
				clearConsole();
```

```
                displayDonationMoney(&MoneyOnce); // Here, the function that displays money have donated once is invoked by passing address of the MoneyOnce Struct.
                break;
            }
            else if (donationTime == "2")
            {
                MoneyMonthly MoneyMonthly;
                addMoney(MoneyMonthly);
                clearConsole();
                cardDonation(cardRow, cardColumns); // Call the card donation function again.
                clearConsole();
                displayDonationMoney(&MoneyMonthly); // Here, the function that displays money have donated monthly is invoked by passing address of the MoneyMonthly Struct.
                break;
            }
            else
            {
                std::cout << "You entered wrong donation time. Please just choose one of the two options mentioned.\n\n";
            }
        }

    }

    else
    {
        std::cout << "You entered wrong donation type. Please just choose one of the two options mentioned.\n \n";
        goto againDonation;
    }

}
```

FLOWCHART

#include <iostream> , #include <sstream> , #include <string>

Necessary libraries were added at the beginning of the project.

struct PersonalDonor

The structure was created for the personal Donor information.

struct CorporateDonor

The structure was created for the corporate Donor information.

struct Date

The structure was created for the date information.

struct Food

The structure was created for the food information.

struct MoneyOnce

The structure was created for a one-time donation.

struct MoneyMonthly

The structure was created for monthly donations.

#include <cstdlib>

Added cstdlib library for void clearConsole() function to work.

void clearConsole()

This function was created to clear the screen.

The sentence (If you are a personal donor please press 1 or a corporate donor press 2) was printed on the screen.

The user is prompted to choose either a personal donor or a corporate donor.

donortype == "1"

If the user selects 1, it goes to the addPersonalDonor(donor) function. In this function, the donor's name, surname, e-mail and phone number are requested to be entered. Then go to the clearConsole() function and the screen is cleared.

donortype == "2"

If the user selects 2, it goes to the addCorporateDonor(donor) function. In this function, the corporation's name, e-mail,corporation's address,corporation's fax number and phone number are requested to be entered. Then go to the clearConsole() function and the screen is cleared.

When Wrong Entry

The sentence (You entered wrong donor type. Please just choose one of the two options mentioned.If you are a personal donor please press 1 or a corporate donor press 2:) was printed on the screen. The user selects it again.

The sentence (Will you donate food or money? If you donate food press 1 or money press 2 :) was printed on the screen.

The user is prompted to select the donation type as food or money.

donationtype == "1"

If the user selects 1, it goes to the addFood(food) function. In this function, the country where the food is donated is asked (it is asked to choose between 1 and 6, if the wrong choice is made, the question is asked again),to select the type of foodgoing to the addTypeofthefooditems(food.foodType) function. In this function (between A and M), a choice is requested, if the wrong choice is made, the question is asked again.If M is selected, going to addFood(selection) functionand the donor is asked to enter the type of food. It is requested to enter the name and amount of the food. It is requested to enter the last expiration day, month and year of the food (if the entered date is a past date, date information is requested again.)It is requested to enter the day, month and year of the date the food will be delivered (if the entered date is a past date or a date after the expiry date, the delivery date information is requested again), added ctime library and time functions so that the time information is compared correctly.. The delivery date is printed on the last screen. Go to the display Donation Food(& food) function and print how much and where the food is donated to the screen.

donationtype == "2"

The sentence (Will you donate money once or monthly? for once press 1; for monthly press 2:) was printed on the screen.

donationtime == "1"

Go to the addMoney(MoneyOnce) function; The country and currency in which the money is donated is selected, and if the wrong choice is made, questions are asked again. The amount of money to be donated is requested. Then go to the carddonation(cardrow, cardcolumns) function; The donor is asked to enter the card number, expiration date (month and year) and CVV code of the card to which the donor will pay, and the donor is thanked for the information entered. Then go to the displayDonationMoney(&MoneyOnce) function and the donated amount and where it was donated are output to the screen.

donationtime == "2"

Go to the addMoney(MoneyMonthly) function; The country and currency in which the money is donated is selected, and if the wrong choice is made, questions are asked again. The amount of money to be donated is requested and the day of the month (between 1-28) the donation is determined. Then go to the carddonation(cardrow, cardcolumns) function; The donor is asked to enter the card number, expiration date (month and year) and CVV code of the card to which the donor will pay, and the donor is thanked for the information entered. Then go to the displayDonationMoney(&MoneyMonthly) function and print the donated amount and where it was donated from.

When Wrong Entry

The sentence You entered wrong donation time. Please just choose one of the two options mentioned.) was printed on the screen. The user selects it again.

Name of the member	Variables/Functions/ Files/Structures	Data type	Return type	Arguments	Description
Emine Hatun Çakmak	`PersonalDonor`	structures	none	none	The structure that holds together information about personal donor.
	`lastName`	string	none	none	The variable holding the last name of personal donor.
	`Name`	string	none	none	The variable holding the name of personal donor.
	`email`	string	none	none	The variable holding the email of personal donor.
	`phone`	string	none	none	The variable holding the phone of personal donor.
	`CorporateDonor`	structures	none	none	The structure that holds together information about corporate donor.
	`name`	string	none	none	The variable holding the name of corporate donor.
	`email`	string	none	none	The variable holding the email of corporate donor.
	`phone`	string	none	none	The variable holding the phone of corporate donor.
	`address`	string	none	none	The variable holding the address of corporate donor.
	`fax`	string	none	none	The variable holding the fax of corporate donor.
	`clearConsole()`	Functions	void	none	This function is used to clean the console for a good look

	addPersonalDonor(PersonalDonor& donor)	Functions	void	donor	this function is used to get the information of personal donor.
	addCorporateDonor(CorporateDonor& donor)	Functions	void	donor	this function is used to get the information of corporate donor.
	donortype	string	none	none	this variable is used to select which functions called.
	ignore()	istream	none	none	Ignoring any characters in the input buffer
	get()	int	none	none	Waiting for the user to press enter
	system("cls")	it does not have a specific data type.	int	cls	system("cls")
	Date	structures	none	none	The structure that holds together information about the delivery date and expiry date.
	expireDay	int	none	none	The variable that holds the day of the entered expiration date.
	expireMonth	int	none	none	A variable that holds the month of the entered expiration date.
	expireYear	int	none	none	A variable that holds the year of the entered expiration date.
	delivery_year	int	none	none	A variable that holds the year of the entered delivery date.
	delivery_month	int	none	none	A variable that holds the month of the entered delivery date.
	delivery_day	int	none	none	A variable that holds the day of the entered delivery date.
	expireDate	Struct of the structure (Type Date)	none	none	It stores information year, day and month of expiration date.

Büşra Basan	`delivery{}`	Struct of the structure (Type Date)	none	none	It stores information year, day and month of delivery date.
	`displayDonationFood()`	none	void	Food* ptr2	This function displays on the screen the amount of the food, the name of the food and the country where is sent that the user has donated by using the address of Food struct argument. In addition, it shows a message on the screen to thank users for their donations.
	`displayDonationMoney()`	none	void	MoneyOnce* ptr2	This function displays on the screen the amount of money the user donates each month, the currency and where the money is sent from by using the address of MoneyOnce struct argument. In addition, it shows a message on the screen to thank users for their donations.
	`displayDonationMoney()`	none	void	MoneyMonthly* ptr2	This function displays the quantity of the money with currency and the country where is sent that the user will donate money monthly and it also shows the day of each month that the user donates. In addition, it indicates a message on the screen to thank users for their donations.
	`ptr2->amountFood`	pointer	none	none	It accesses the address of the member of the Food struct and call the data inside the amountFood and uses in the displayDonationFood() function to show the amount of the food.

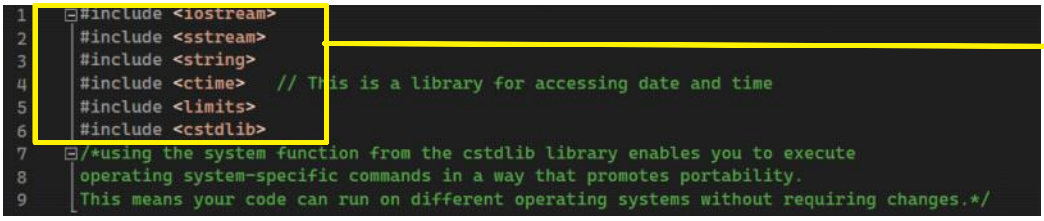
	`ptr2->selectedCountry`	pointer	none	none	It accesses the address of the member of the Food struct and call the data inside the selectedCountr and uses in the displayDonationFood() and displayDonationMoney() functions to shows the user where is sent .
	`ptr2->quantityOfTheDonation`	pointer	none	none	It accesses the address of the member of the Food struct and call the data inside the quantityOfTheDonation and uses in the displayDonationMoney () function to show the quantity of the donation the user has donated.
	`ptr2->monthDay`	pointer	none	none	It accesses the address of the member of the MoneyMonthly struct and call the data inside the monthDay and uses the displayDonationFood() function to show which day the user donates every month on the screen.
	`ptr2->selectedCurrency`	pointer	none	none	It accesses the address of the member of the MoneyMonthly struct and call the data inside the monthDay and uses the displayDonationMoney () functions to show the currency on the screen.
	`Food`	structures	none	none	The structure that holds together information about food.
	`donationCountry[6]`	array of structure (Type Food)	none	none	It stores records of six countries.
	`selectedCountry`	string	none	none	The variable holding the selected country.

Zeynep Can	foodType	string	none	none	Variable that stores the type of food entered or selected.
	foodName	string	none	none	Variable that stores the name of food entered
	amountFood	string	none	none	Variable that stores the amount of food entered.
	addFood()	none	void	std::string &select	Function that allows the type of food to be written by the user.
	addTypeofthefooditems()	none	void	std::string & selection	Function that allows the type of food to be selected by the user.
	addFood()	none	void	Food&food	In this function, first, the country where the food is donated is selected. Then, the function goes to the "Typeofthefooditems()" function. After that, the donor enters the name of the food. Then, the donor enters the quantity of the food to be sent. Next, the donor enters the expiration date of the food and the date when the food will be delivered. In our function, the date information is compared, and the donor is prompted to enter the correct date information until the valid date is entered.
	ignore()	istream	none	none	Ignoring any characters in the input buffer

	choice	int	none	none	Stores the selected country variable.
	todayDay	int	none	none	The variable stores the current day information.
	todayMonth	int	none	none	The variable stores the current month information.
	todayYear	int	none	none	The variable stores the current year information.
	localtime_s()	none	void	t	Converts the time to the local time zone.
	t	time_t	none	none	It stores the time information in the system.
	today	tm	none	none	It stores the time information in the system.
	MoneyOnce	structures	none	none	This structure holds the necessary information for one-time money donation transaction.
	MoneyMonthly	structures	none	none	This struct holds the necessary information for a recurring fundraiser once a month.
	donationCountry[6]	arrays of structure (type MoneyOnce)	none	none	It stores records of six countries.
	selectedCountry	string	none	none	The variable holding the selected country.
	quantityOfTheDonation{}	int	none	none	This variable defines the quantity of the money.

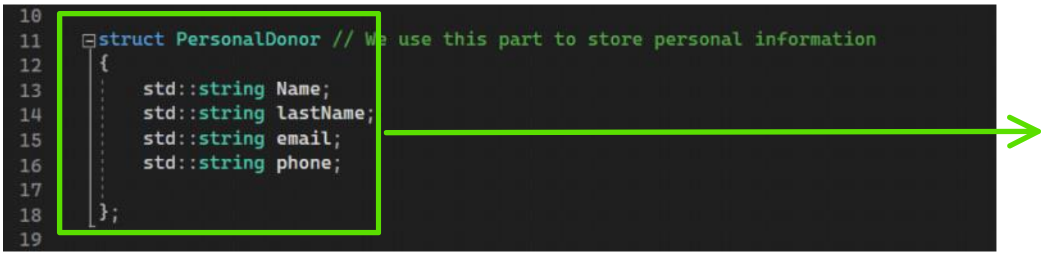
Göksu Kesen	donationCurrency[6]	string	none	none	It stores records of six currency.
	selectedCurrency	string	none	none	The variable holding the selected currency.
	monthDay	int	none	none	this variable defines which day of each month the user will donate.
	addMoney	void	none	Donation Once	The function retrieves the donor country, donation amount and currency selected by the user.
	choice	int	none	none	This variable is defined for country selection in the one-time donation function..
	choice2	int	none	none	This variable is defined for currency selection in the one-time donation function.
	addMoney	void	none	Donation Monthly	This function adds a money donation transaction based on the "MoneyMonthly" struct.The function retrieves the donor country, donation amount and currency selected by the user.
	choice3	int	none	none	This variable is defined for country selection in the monthly donation function.
	choice4	int	none	none	This variable is defined for currency selection in the monthly donation function.
	donationTime	string	none	none	This function is defined to ask the user whether to make a one-time or monthly donation.
	donationType	string	none	none	This function is defined to be able to ask the user whether to donate money or food.
	cardNumber	functions	bool	none	Function to check if the card number is valid.
	cardDonation	functions	void	none	Function to process card donation.

Kadir Ocak	`expirationDate`	string	none	none	To take expiration date of credit card.
	`cvvCode`	string	none	none	To take cvv code from the user.
	`cardRow`	int	none	none	A part of card number row.
	`cardNumber`	int	none	none	Prompt the user to enter card numbers.
	`cardColumns`	int	none	none	A row of cards are cardColums.
	`number`	int	none	none	To check if the card number is valid.
	`cardDonation{}`	functions	void	none	Call the card donation funciton.



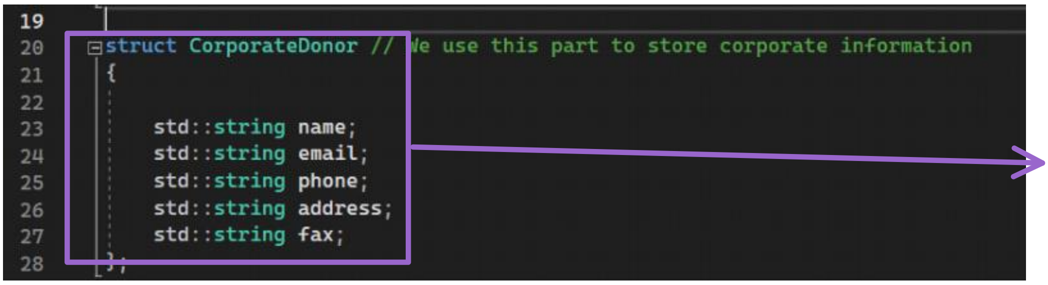
```
#include <iostream>
#include <sstream>
#include <string>
#include <ctime>    // This is a library for accessing date and time
#include <limits>
#include <cstdlib>
/*using the system function from the cstdlib library enables you to execute
operating system-specific commands in a way that promotes portability.
This means your code can run on different operating systems without requiring changes.*/
```

These are required libraries for our project.



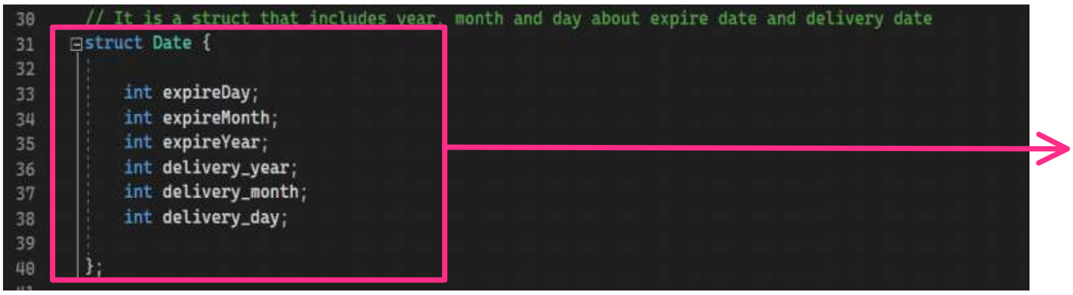
```

struct PersonalDonor // We use this part to store personal information
{
    std::string Name;
    std::string lastName;
    std::string email;
    std::string phone;

};

```

This structure holds data about personal donor together.



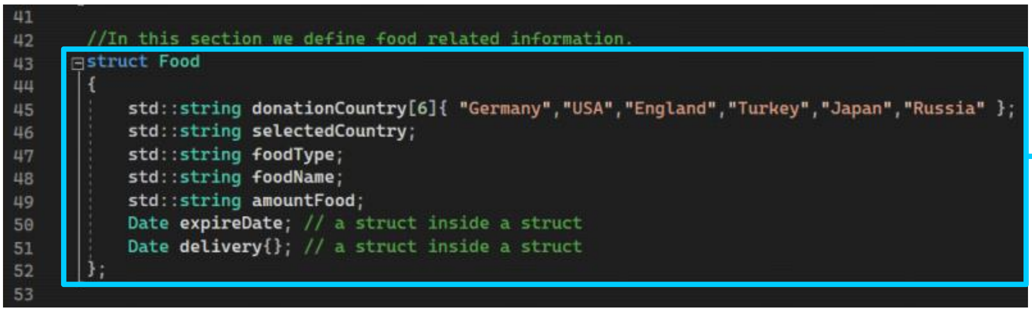
```

struct CorporateDonor // We use this part to store corporate information
{

    std::string name;
    std::string email;
    std::string phone;
    std::string address;
    std::string fax;
};
```

This structure holds data about corporate donor together.



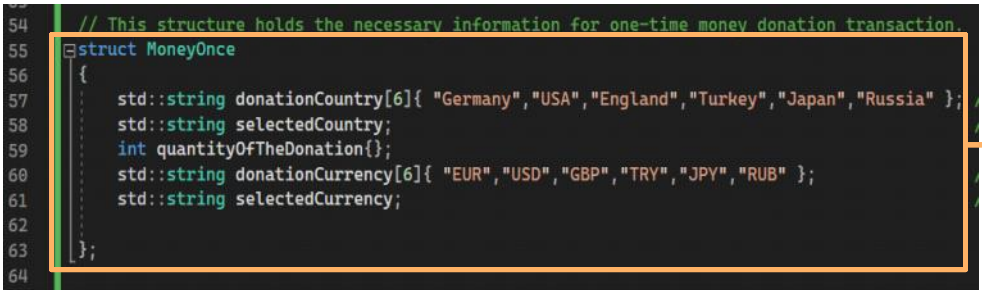
```
// It is a struct that includes year, month and day about expire date and delivery date
struct Date {

    int expireDay;
    int expireMonth;
    int expireYear;
    int delivery_year;
    int delivery_month;
    int delivery_day;

};
```

This structure holds year, month and day ,which are variables related to expiration date and delivery date.



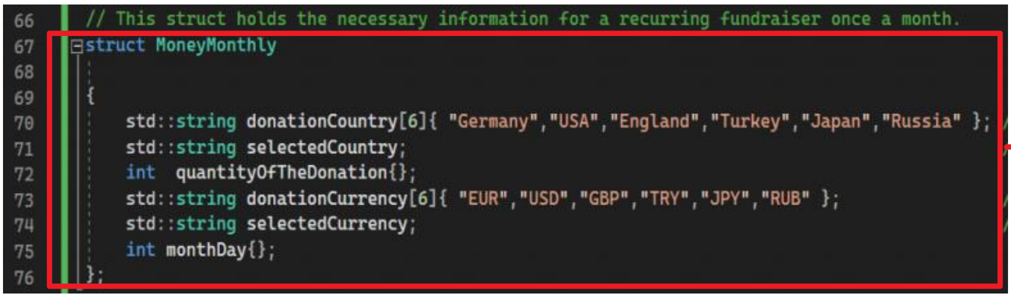
```

//In this section we define food related information.
struct Food
{
    std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" };
    std::string selectedCountry;
    std::string foodType;
    std::string foodName;
    std::string amountFood;
    Date expireDate; // a struct inside a struct
    Date delivery{}; // a struct inside a struct
};

```

This structure holds data about food such as food name, quantity, type and country and date structure.



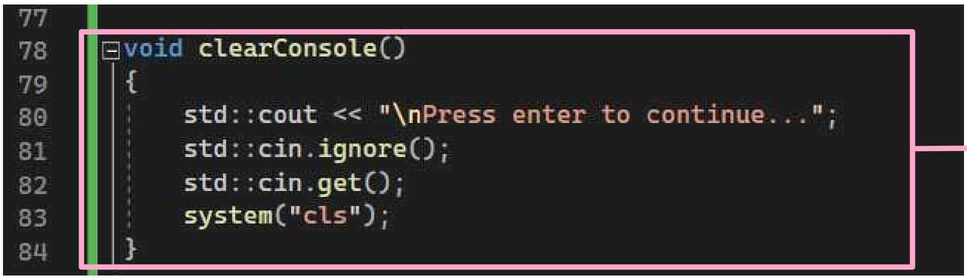
```cpp
// This structure holds the necessary information for one-time money donation transaction.
struct MoneyOnce
{
    std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" };
    std::string selectedCountry;
    int quantityOfTheDonation{};
    std::string donationCurrency[6]{ "EUR","USD","GBP","TRY","JPY","RUB" };
    std::string selectedCurrency;

};
```

This structure holds data about money that the user has donated once.



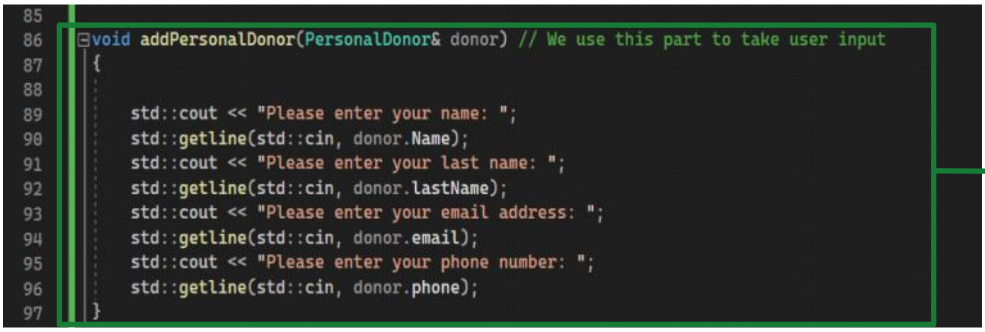
```cpp
// This struct holds the necessary information for a recurring fundraiser once a month.
struct MoneyMonthly

{
    std::string donationCountry[6]{ "Germany","USA","England","Turkey","Japan","Russia" };
    std::string selectedCountry;
    int  quantityOfTheDonation{};
    std::string donationCurrency[6]{ "EUR","USD","GBP","TRY","JPY","RUB" };
    std::string selectedCurrency;
    int monthDay{};
};
```

This structure holds data about money that the user has donated every month. For instance, country ,quantity, currency and day.



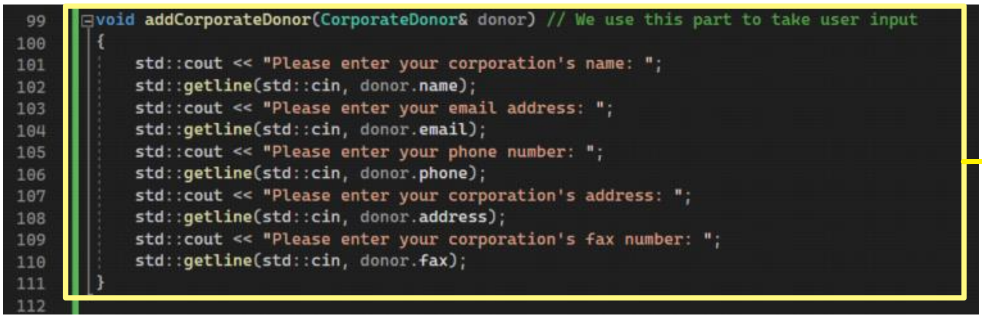
```cpp
void clearConsole()
{
    std::cout << "\nPress enter to continue...";
    std::cin.ignore();
    std::cin.get();
    system("cls");
}
```

It is a function created to clear the console. Thanks to this, a simpler console screen is presented to the user.



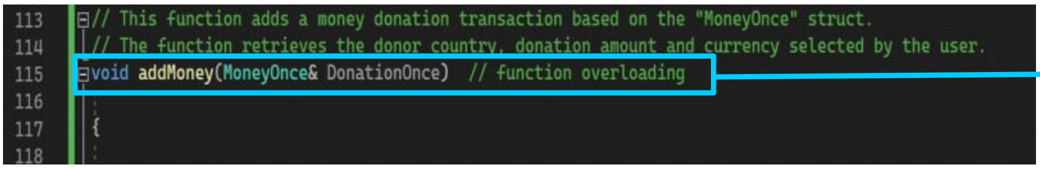
```cpp
void addPersonalDonor(PersonalDonor& donor) // We use this part to take user input
{

    std::cout << "Please enter your name: ";
    std::getline(std::cin, donor.Name);
    std::cout << "Please enter your last name: ";
    std::getline(std::cin, donor.lastName);
    std::cout << "Please enter your email address: ";
    std::getline(std::cin, donor.email);
    std::cout << "Please enter your phone number: ";
    std::getline(std::cin, donor.phone);
}
```

This function takes input related to personal information from the personal user .



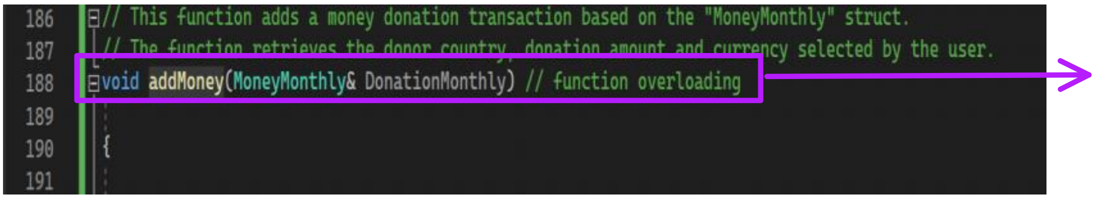
```cpp
void addCorporateDonor(CorporateDonor& donor) // We use this part to take user input
{
    std::cout << "Please enter your corporation's name: ";
    std::getline(std::cin, donor.name);
    std::cout << "Please enter your email address: ";
    std::getline(std::cin, donor.email);
    std::cout << "Please enter your phone number: ";
    std::getline(std::cin, donor.phone);
    std::cout << "Please enter your corporation's address: ";
    std::getline(std::cin, donor.address);
    std::cout << "Please enter your corporation's fax number: ";
    std::getline(std::cin, donor.fax);
}
```

This function takes input related to corporate information from the company.



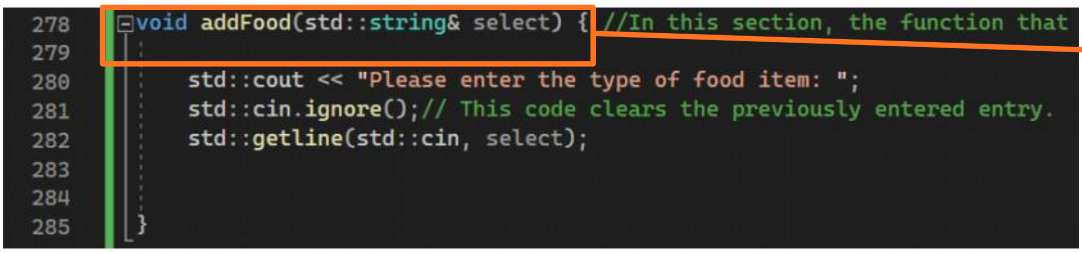
```
// This function adds a money donation transaction based on the "MoneyOnce" struct.
// The function retrieves the donor country, donation amount and currency selected by the user.
void addMoney(MoneyOnce& DonationOnce)  // function overloading

{

```

This function holds the quantity of the money , country, currency if the user has donated money once and goes to the payment part.



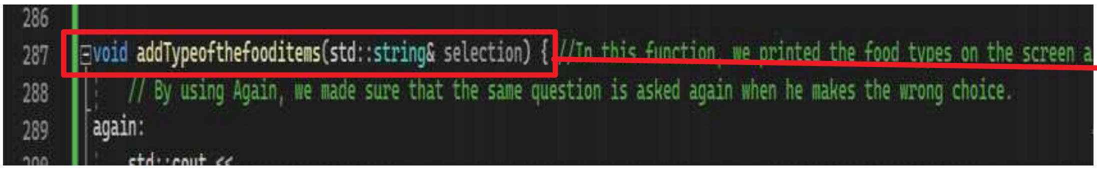
```
// This function adds a money donation transaction based on the "MoneyMonthly" struct.
// The function retrieves the donor country, donation amount and currency selected by the user.
void addMoney(MoneyMonthly& DonationMonthly) // function overloading

{

```

This function holds the quantity of the money , country, currency and date if the user has donated money every month and goes to the payment part.



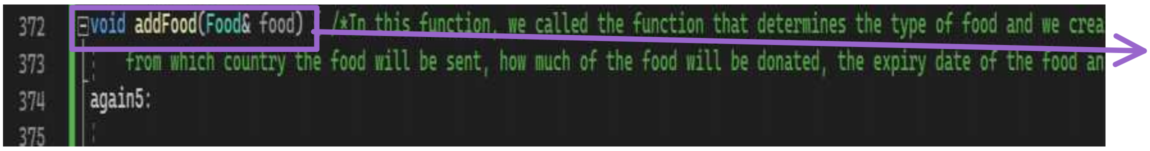
```
void addFood(std::string& select) { //In this section, the function that

    std::cout << "Please enter the type of food item: ";
    std::cin.ignore();// This code clears the previously entered entry.
    std::getline(std::cin, select);


}
```

This function provides the user to enter the type of the food.



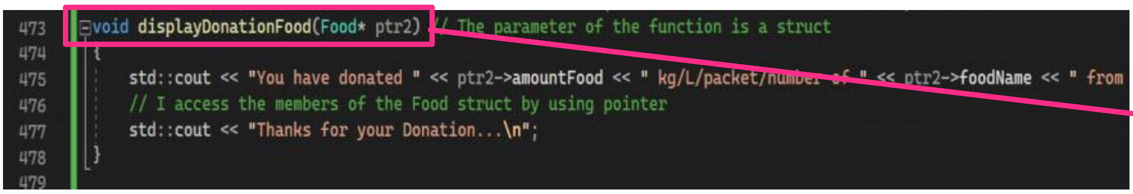
```

void addTypeofthefooditems(std::string& selection) { //In this function, we printed the food types on the screen a
    // By using Again, we made sure that the same question is asked again when he makes the wrong choice.
again:
    std::cout <<
```

This function provides the user to choose the food items if the user wants to donate food.



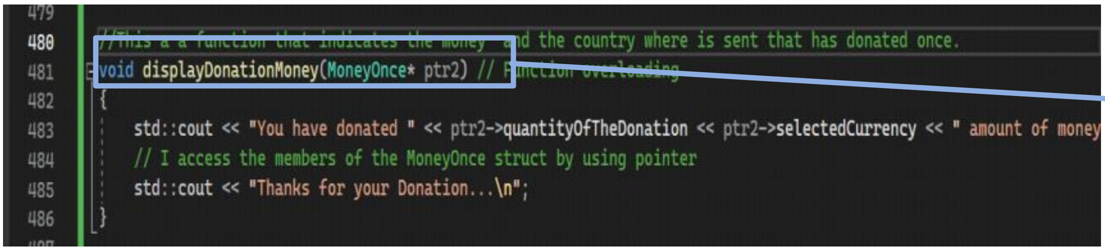
```
void addFood(Food& food) { /*In this function, we called the function that determines the type of food and we crea
    from which country the food will be sent, how much of the food will be donated, the expiry date of the food an
again5:

```

In this function, we called the function that determines the type of food and we created this function to learn the name of the food, from which country the food will be sent, how much of the food will be donated, the expiry date of the food and the delivery date.



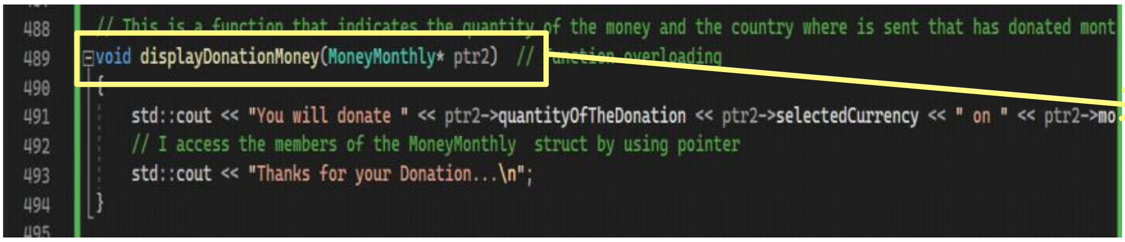
```cpp
void displayDonationFood(Food* ptr2) // The parameter of the function is a struct
{
    std::cout << "You have donated " << ptr2->amountFood << " kg/L/packet/number of " << ptr2->foodName << " from
    // I access the members of the Food struct by using pointer
    std::cout << "Thanks for your Donation...\n";
}
```

This function displays on the screen food name, amount of the food , the country where is sent that have donated if the user has donated food.In addition, it shows a message on the screen to thank users for their donations.



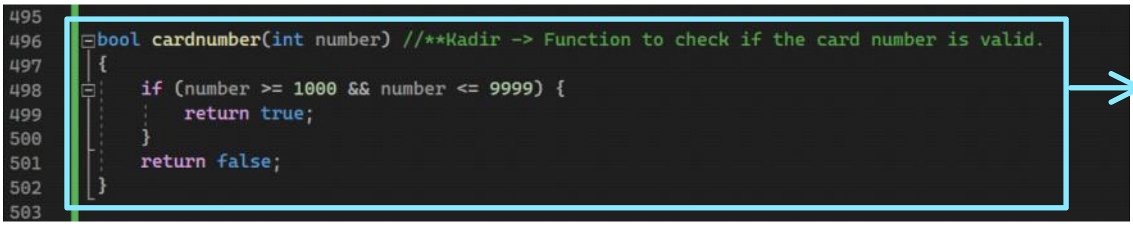
```cpp
//This a a function that indicates the money  and the country where is sent that has donated once.
void displayDonationMoney(MoneyOnce* ptr2) // Function overloading
{
    std::cout << "You have donated " << ptr2->quantityOfTheDonation << ptr2->selectedCurrency << " amount of money
    // I access the members of the MoneyOnce struct by using pointer
    std::cout << "Thanks for your Donation...\n";
}
```

This function displays on the screen the money, currency and the country where is sent if the user has donated money once.In addition, it shows a message on the screen to thank users for their donations.



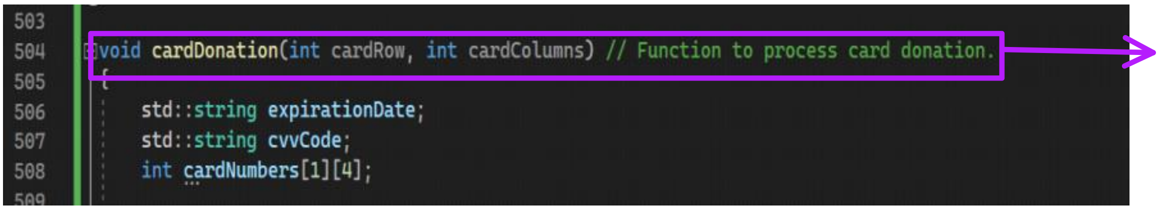
```cpp
// This is a function that indicates the quantity of the money and the country where is sent that has donated mont
void displayDonationMoney(MoneyMonthly* ptr2)  // Function overloading
{
    std::cout << "You will donate " << ptr2->quantityOfTheDonation << ptr2->selectedCurrency << " on " << ptr2->mo
    // I access the members of the MoneyMonthly  struct by using pointer
    std::cout << "Thanks for your Donation...\n";
}
```

This function displays on the screen the money, currency , day and country where is sent if the user has donated money every month. In addition, it shows a message on the screen to thank users for their donations.



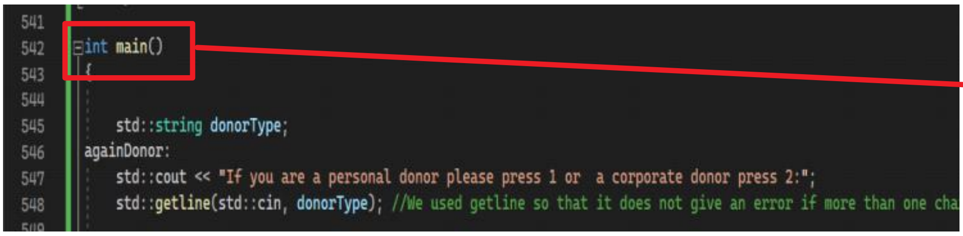
```cpp
bool cardnumber(int number) //**Kadir -> Function to check if the card number is valid.
{
    if (number >= 1000 && number <= 9999) {
        return true;
    }
    return false;
}
```

This function checks the card number whether is valid or not.



```
void cardDonation(int cardRow, int cardColumns) // Function to process card donation.
{
    std::string expirationDate;
    std::string cvvCode;
    int cardNumbers[1][4];
```

This function provides the user to enter her/ him payment information if the user chooses the money for donation.In addition, it shows a message on the screen to thank users for their donations.

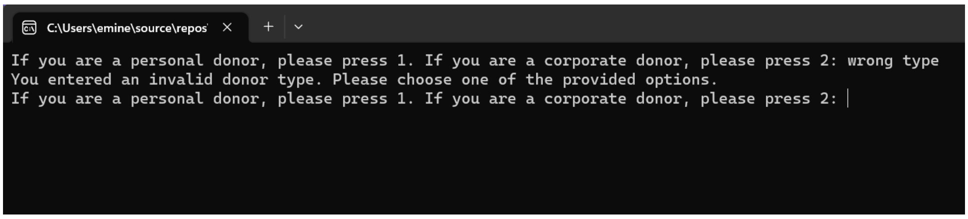


```
int main()
{

    std::string donorType;
againDonor:
    std::cout << "If you are a personal donor please press 1 or  a corporate donor press 2:";
    std::getline(std::cin, donorType); //We used getline so that it does not give an error if more than one cha
```

In the main function, all requried variables are created, all required inputs are entered and all required functions are called.

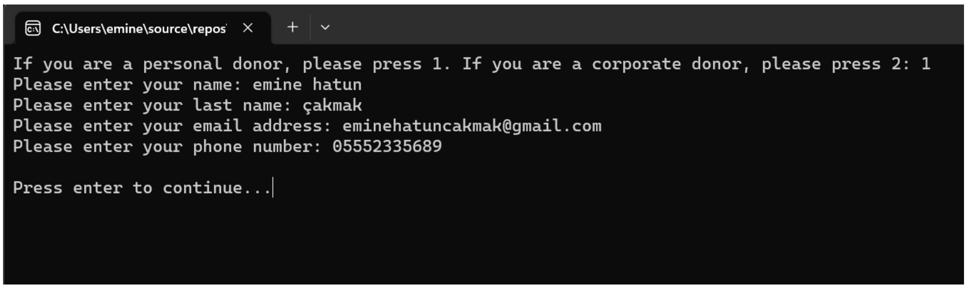
OU TPU T

When you enter a value other than 1 and 2, you will be directed to this page.



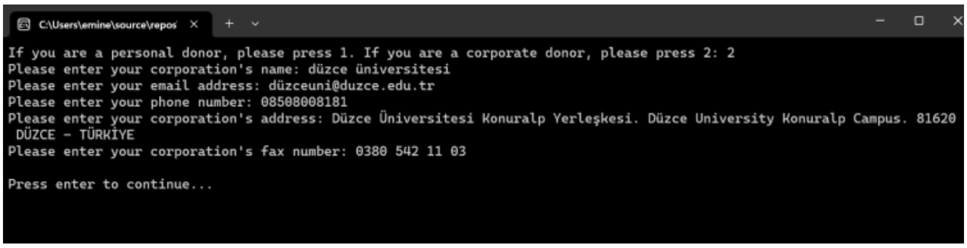
```
If you are a personal donor, please press 1. If you are a corporate donor, please press 2: wrong type
You entered an invalid donor type. Please choose one of the provided options.
If you are a personal donor, please press 1. If you are a corporate donor, please press 2:
```

If you choose 1, you will be directed to the personal donor page.



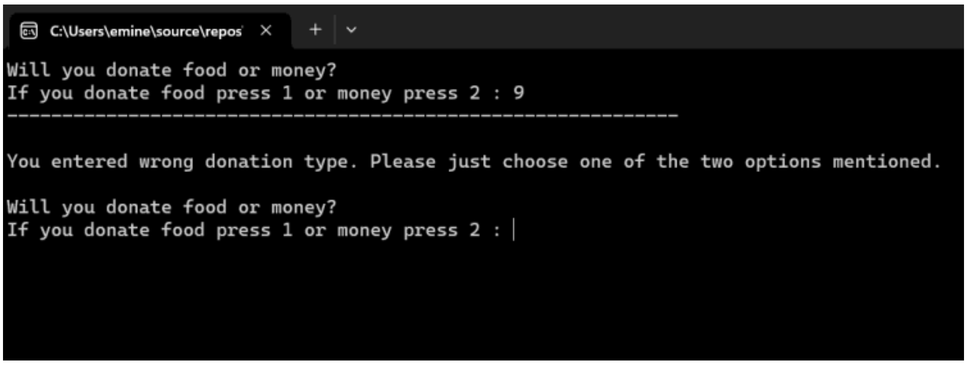
```
If you are a personal donor, please press 1. If you are a corporate donor, please press 2: 1
Please enter your name: emine hatun
Please enter your last name: çakmak
Please enter your email address: eminehatuncakmak@gmail.com
Please enter your phone number: 05552335689

Press enter to continue...
```

If you choose 2, you will be directed to the corporate page.



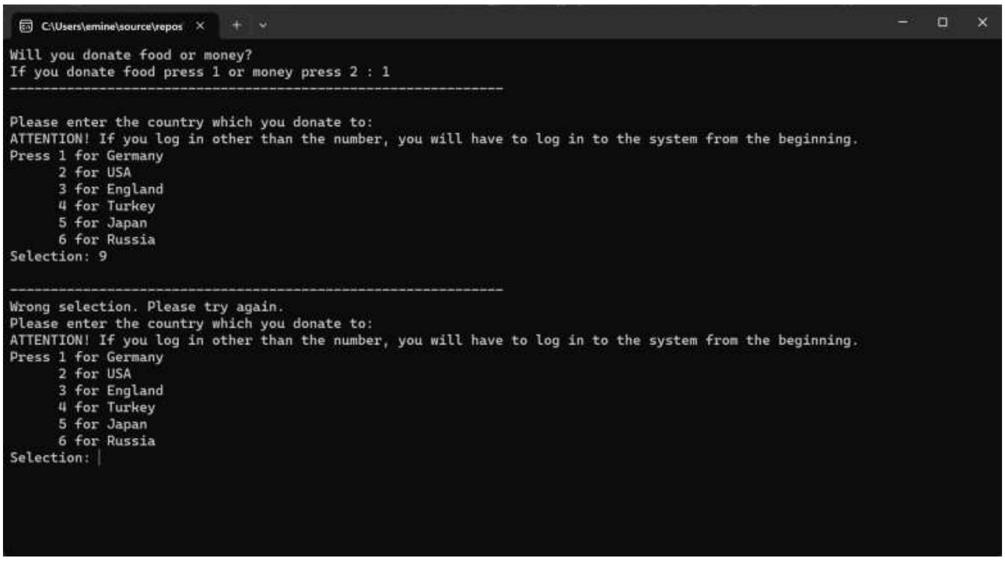
```
If you are a personal donor, please press 1. If you are a corporate donor, please press 2: 2
Please enter your corporation's name: düzce üniversitesi
Please enter your email address: düzceuni@duzce.edu.tr
Please enter your phone number: 08508008181
Please enter your corporation's address: Düzce Üniversitesi Konuralp Yerleşkesi. Düzce University Konuralp Campus. 81620
 DÜZCE - TÜRKİYE
Please enter your corporation's fax number: 0380 542 11 03

Press enter to continue...
```

console is cleared and this page is switched. If you choose anything other than 1 or 2 in donation type, you will be redirected to this page.



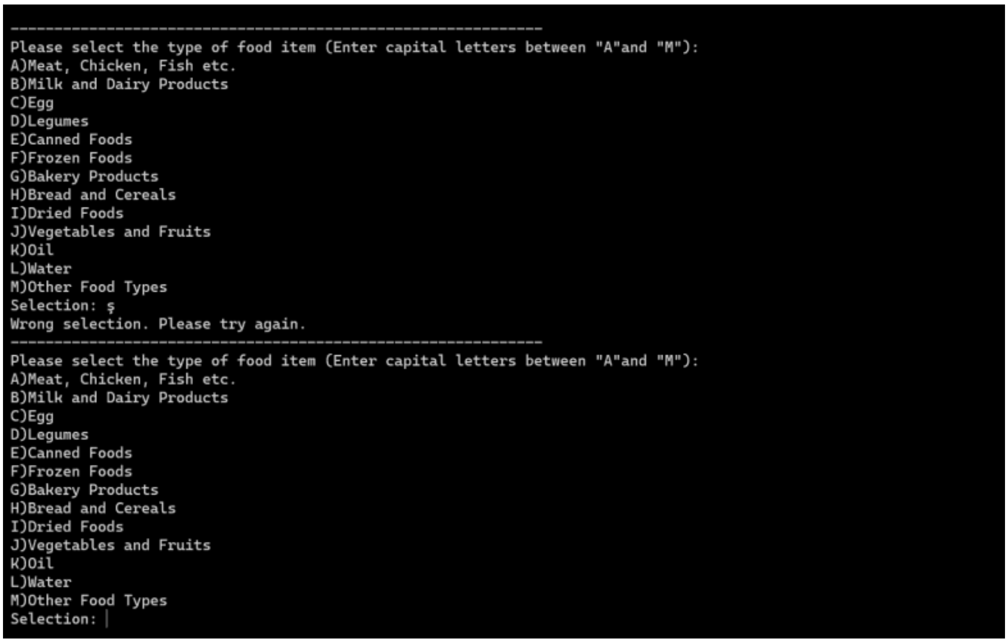
```
Will you donate food or money?
If you donate food press 1 or money press 2 : 9
------------------------------------------------------------

You entered wrong donation type. Please just choose one of the two options mentioned.

Will you donate food or money?
If you donate food press 1 or money press 2 :
```

If you choose 1, you will be directed to the food donation page.
There is a list on this page asking which country you donated from. If you enter a number other than the members of this list, you will see a text asking you to enter again.



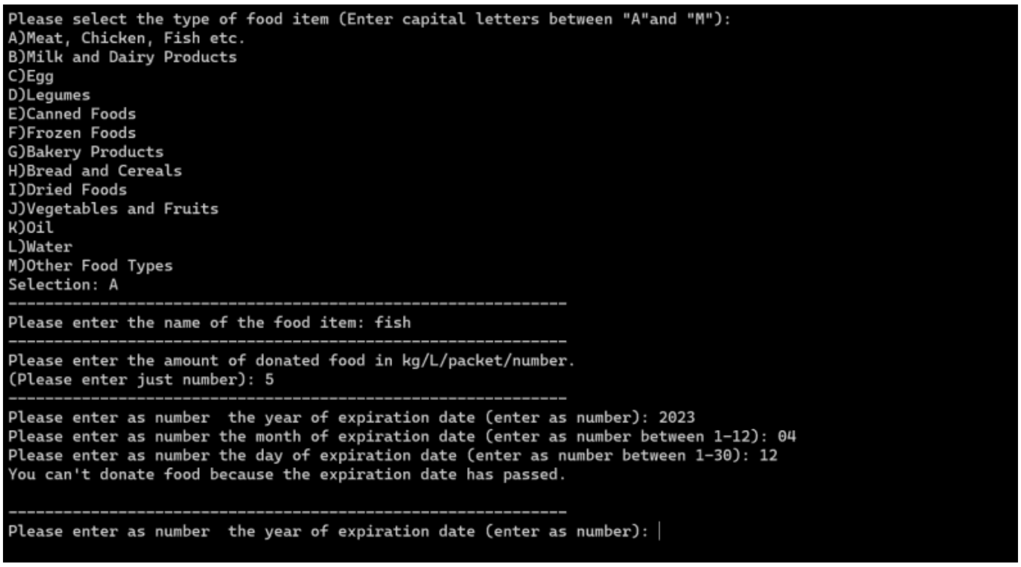
```
Will you donate food or money?
If you donate food press 1 or money press 2 : 1
-------------------------------------------------------------

Please enter the country which you donate to:
ATTENTION! If you log in other than the number, you will have to log in to the system from the beginning.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection: 9

-------------------------------------------------------------
Wrong selection. Please try again.
Please enter the country which you donate to:
ATTENTION! If you log in other than the number, you will have to log in to the system from the beginning.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection:
```

When you enter the correct value on the previous page, you will be directed to the food item page. If you select anything other than those defined here, you will see this warning.



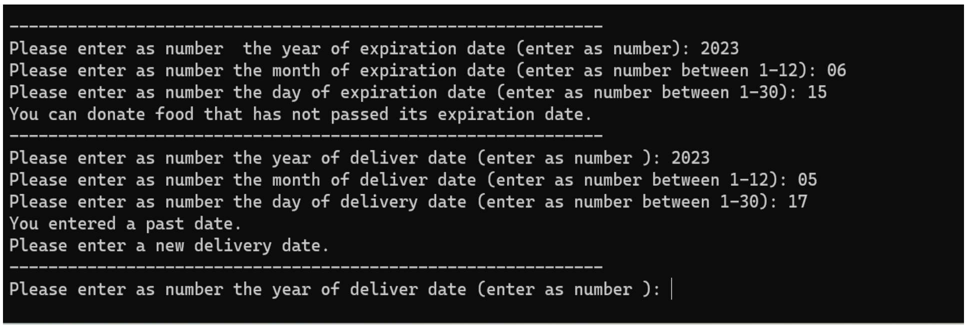
```
-------------------------------------------------------------
Please select the type of food item (Enter capital letters between "A"and "M"):
A)Meat, Chicken, Fish etc.
B)Milk and Dairy Products
C)Egg
D)Legumes
E)Canned Foods
F)Frozen Foods
G)Bakery Products
H)Bread and Cereals
I)Dried Foods
J)Vegetables and Fruits
K)Oil
L)Water
M)Other Food Types
Selection: ş
Wrong selection. Please try again.
-------------------------------------------------------------
Please select the type of food item (Enter capital letters between "A"and "M"):
A)Meat, Chicken, Fish etc.
B)Milk and Dairy Products
C)Egg
D)Legumes
E)Canned Foods
F)Frozen Foods
G)Bakery Products
H)Bread and Cereals
I)Dried Foods
J)Vegetables and Fruits
K)Oil
L)Water
M)Other Food Types
Selection:
```

When you enter the required information correctly, this time you will see an entry asking for the expiry date of the product. if you enter a past date here you will get this warning.



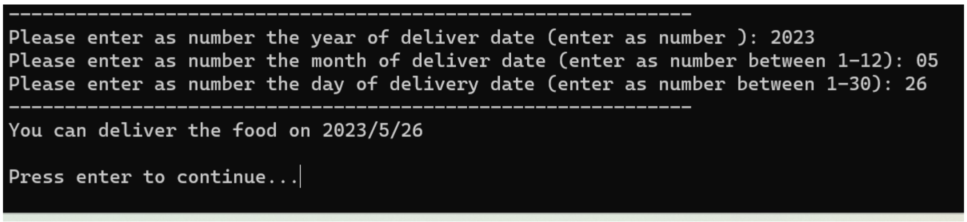
```
Please select the type of food item (Enter capital letters between "A"and "M"):
A)Meat, Chicken, Fish etc.
B)Milk and Dairy Products
C)Egg
D)Legumes
E)Canned Foods
F)Frozen Foods
G)Bakery Products
H)Bread and Cereals
I)Dried Foods
J)Vegetables and Fruits
K)Oil
L)Water
M)Other Food Types
Selection: A
------------------------------------------------------------
Please enter the name of the food item: fish
------------------------------------------------------------
Please enter the amount of donated food in kg/L/packet/number.
(Please enter just number): 5
------------------------------------------------------------
Please enter as number  the year of expiration date (enter as number): 2023
Please enter as number the month of expiration date (enter as number between 1-12): 04
Please enter as number the day of expiration date (enter as number between 1-30): 12
You can't donate food because the expiration date has passed.

------------------------------------------------------------
Please enter as number  the year of expiration date (enter as number): |
```

After the expiry date, the question asked was on which date you will send the products you will send. If you enter a past date here you will get this warning.



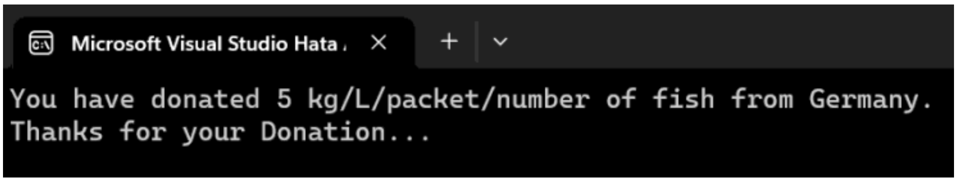
```
------------------------------------------------------------
Please enter as number  the year of expiration date (enter as number): 2023
Please enter as number the month of expiration date (enter as number between 1-12): 06
Please enter as number the day of expiration date (enter as number between 1-30): 15
You can donate food that has not passed its expiration date.
------------------------------------------------------------
Please enter as number the year of deliver date (enter as number ): 2023
Please enter as number the month of deliver date (enter as number between 1-12): 05
Please enter as number the day of delivery date (enter as number between 1-30): 17
You entered a past date.
Please enter a new delivery date.
------------------------------------------------------------
Please enter as number the year of deliver date (enter as number ): |
```

If you enter the correct dates you will have a view like this.



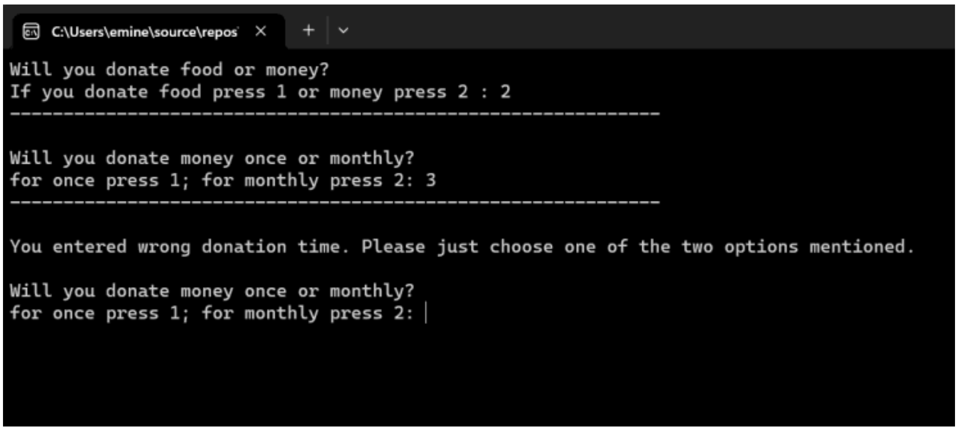
```
------------------------------------------------------------
Please enter as number the year of deliver date (enter as number ): 2023
Please enter as number the month of deliver date (enter as number between 1-12): 05
Please enter as number the day of delivery date (enter as number between 1-30): 26
------------------------------------------------------------
You can deliver the food on 2023/5/26

Press enter to continue...|
```

This page is an display of your donation.



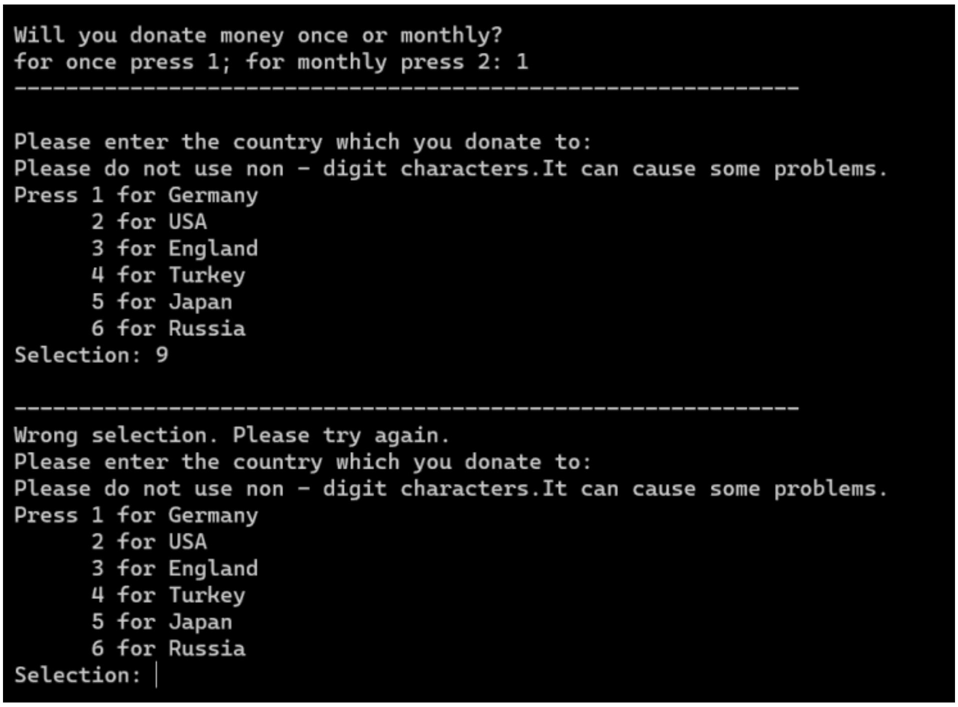
```
You have donated 5 kg/L/packet/number of fish from Germany.
Thanks for your Donation...
```

If you choose number 2 in the donation type, you will be directed to the money donation section where you will be presented with two options to donate money, as a one-time donation or a monthly donation, if you enter a value other than 1 or 2 here, you will see this warning



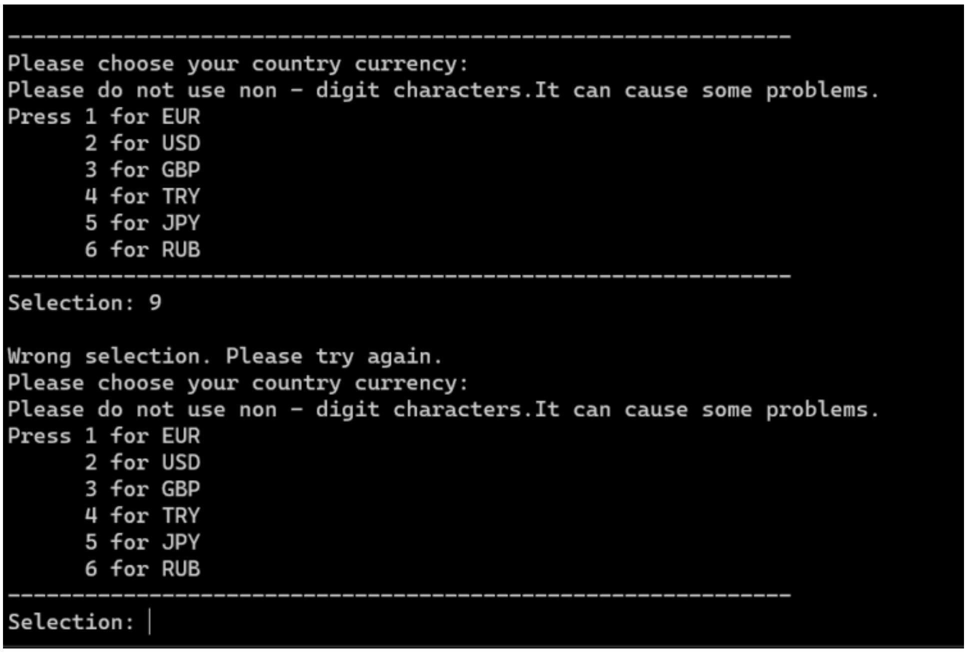
```
Will you donate food or money?
If you donate food press 1 or money press 2 : 2
------------------------------------------------------------

Will you donate money once or monthly?
for once press 1; for monthly press 2: 3
------------------------------------------------------------

You entered wrong donation time. Please just choose one of the two options mentioned.

Will you donate money once or monthly?
for once press 1; for monthly press 2:
```

When we choose the once donation part here, it asks us which country we chose from. It repeats the question when we make a wrong entry.



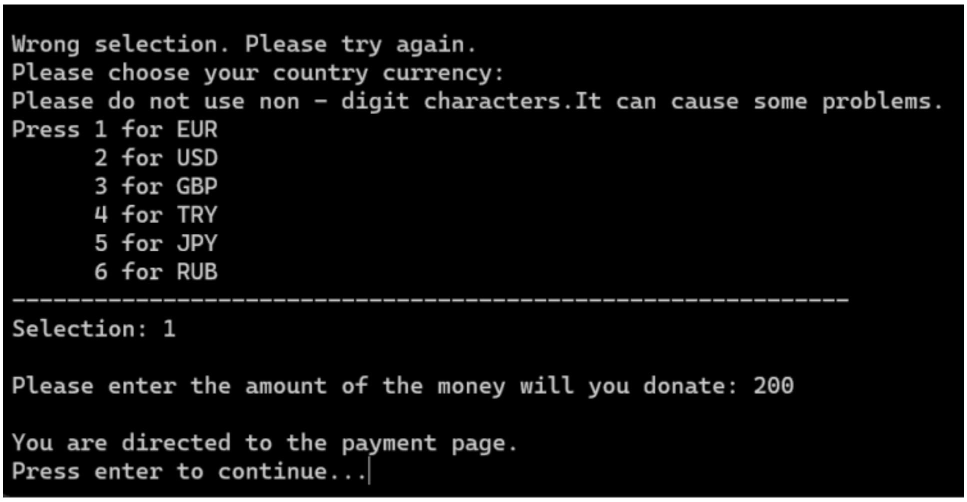
```
Will you donate money once or monthly?
for once press 1; for monthly press 2: 1
------------------------------------------------------------

Please enter the country which you donate to:
Please do not use non - digit characters.It can cause some problems.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection: 9

------------------------------------------------------------
Wrong selection. Please try again.
Please enter the country which you donate to:
Please do not use non - digit characters.It can cause some problems.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection:
```

In the previous section, we encounter a section that asks for currency after the correct input is taken. repeats the same question when wrong input is received.



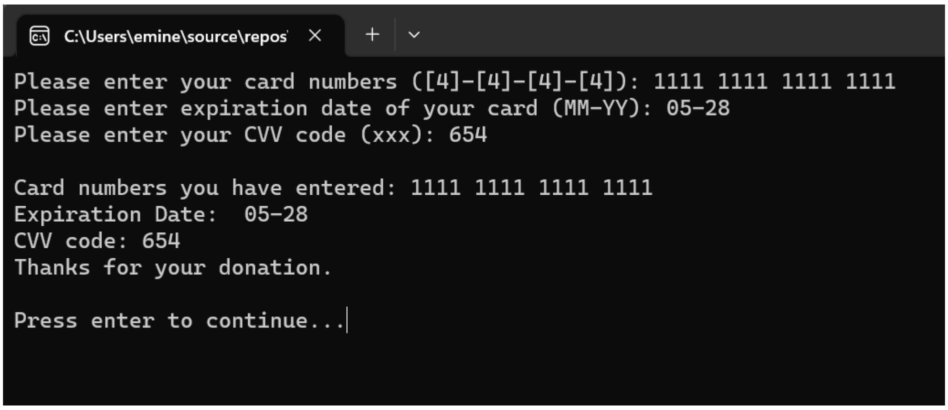
```
------------------------------------------------------------
Please choose your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
------------------------------------------------------------
Selection: 9

Wrong selection. Please try again.
Please choose your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
------------------------------------------------------------
Selection: 
```

After correct input you will reach this page.



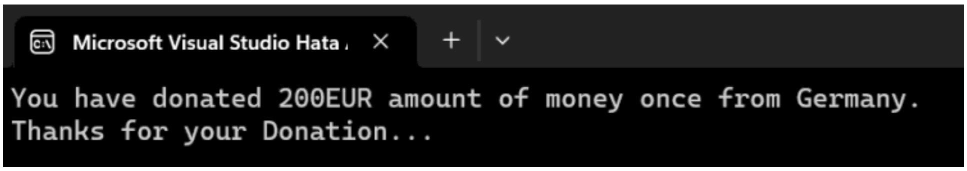
```
Wrong selection. Please try again.
Please choose your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
------------------------------------------------------------
Selection: 1

Please enter the amount of the money will you donate: 200

You are directed to the payment page.
Press enter to continue...
```

When you are redirected to the payment page, you will be prompted to enter this information. you will get an output based on this information



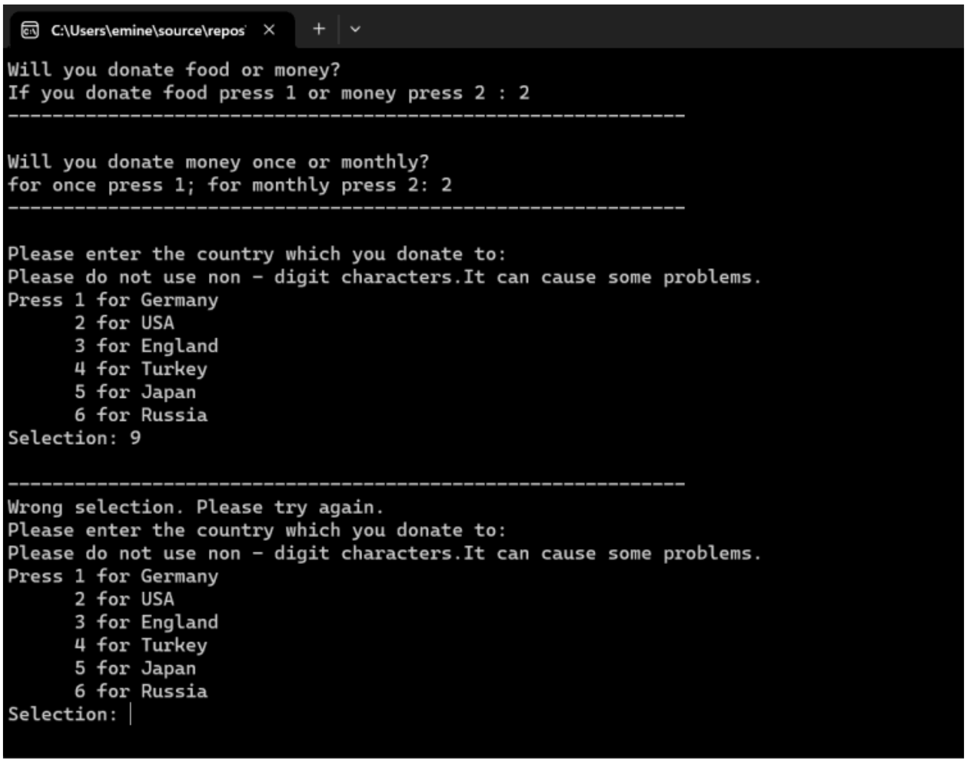
```
Please enter your card numbers ([4]-[4]-[4]-[4]): 1111 1111 1111 1111
Please enter expiration date of your card (MM-YY): 05-28
Please enter your CVV code (xxx): 654

Card numbers you have entered: 1111 1111 1111 1111
Expiration Date:  05-28
CVV code: 654
Thanks for your donation.

Press enter to continue...
```

This page is an display of your donation.



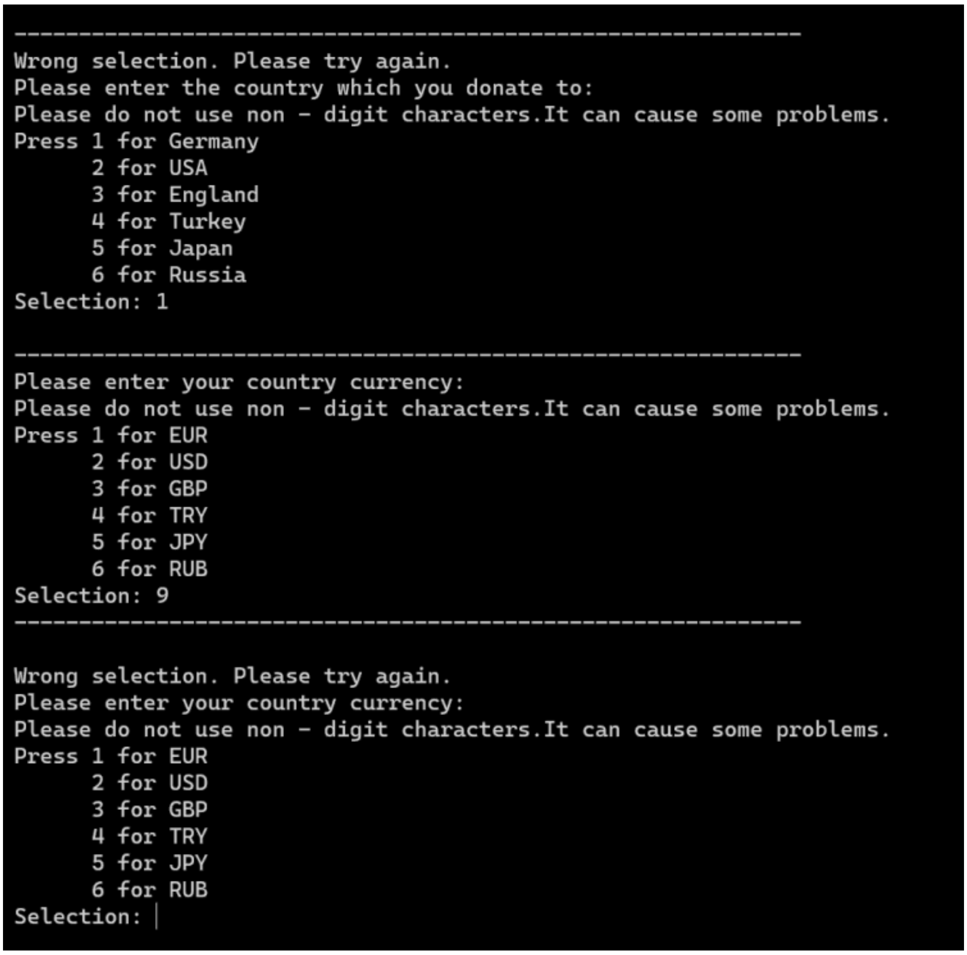
```
You have donated 200EUR amount of money once from Germany.
Thanks for your Donation...
```

when we select 2 for monthly payment we will see this page. In case of incorrect entry, the question is repeated.



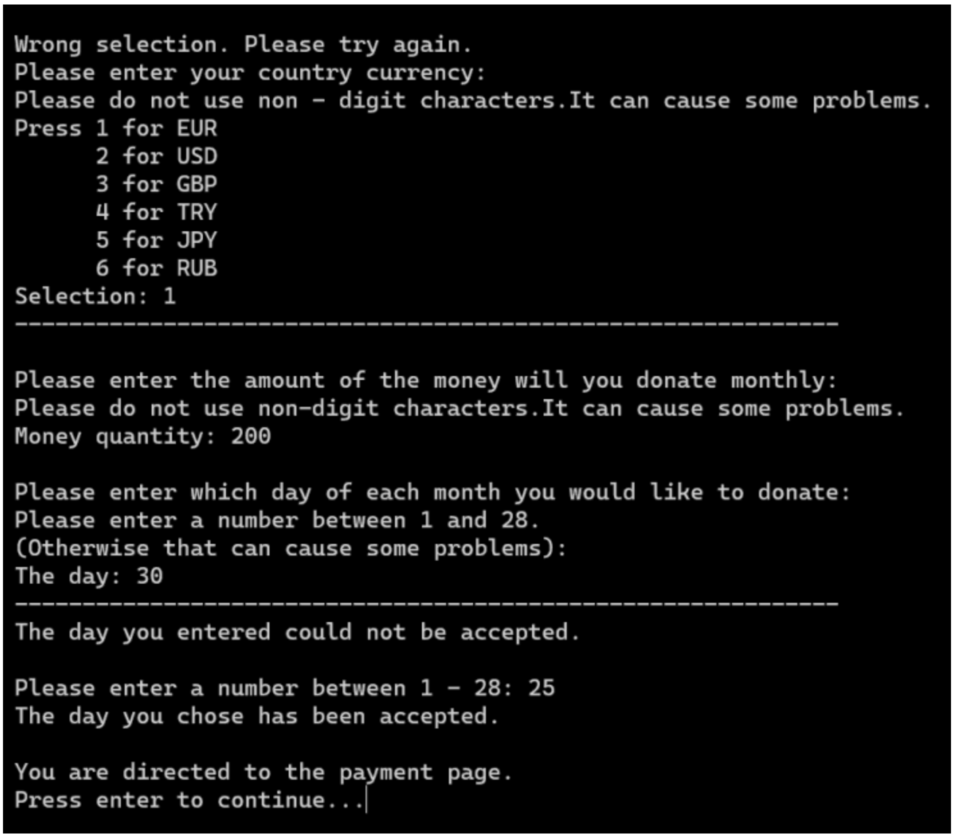
```
Will you donate food or money?
If you donate food press 1 or money press 2 : 2
------------------------------------------------------------

Will you donate money once or monthly?
for once press 1; for monthly press 2: 2
------------------------------------------------------------

Please enter the country which you donate to:
Please do not use non - digit characters.It can cause some problems.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection: 9

------------------------------------------------------------
Wrong selection. Please try again.
Please enter the country which you donate to:
Please do not use non - digit characters.It can cause some problems.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection:
```

After the correct entry for the country, the same procedure is applied for the currency.



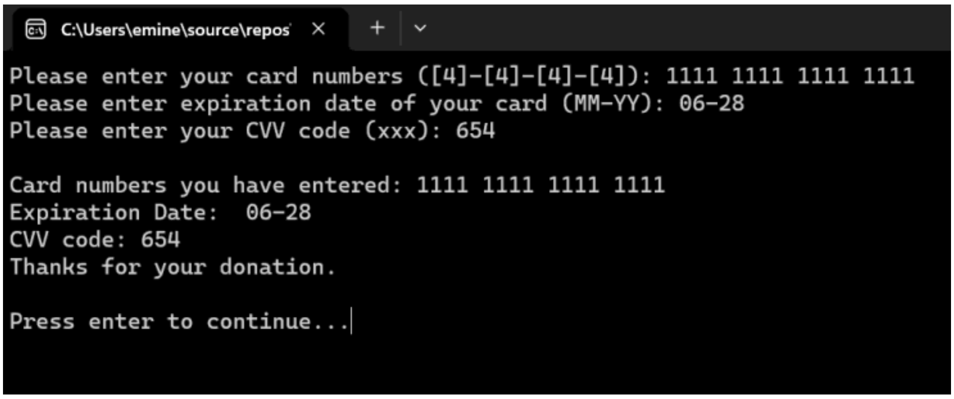
```
-------------------------------------------------------------
Wrong selection. Please try again.
Please enter the country which you donate to:
Please do not use non - digit characters.It can cause some problems.
Press 1 for Germany
      2 for USA
      3 for England
      4 for Turkey
      5 for Japan
      6 for Russia
Selection: 1

-------------------------------------------------------------
Please enter your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
Selection: 9
-------------------------------------------------------------

Wrong selection. Please try again.
Please enter your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
Selection:
```

Here you can get information on which day of each month the donation will be made.



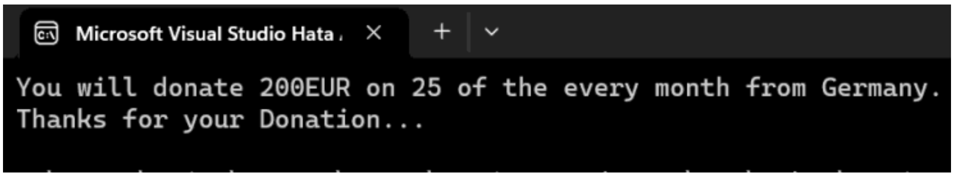
```
Wrong selection. Please try again.
Please enter your country currency:
Please do not use non - digit characters.It can cause some problems.
Press 1 for EUR
      2 for USD
      3 for GBP
      4 for TRY
      5 for JPY
      6 for RUB
Selection: 1
-------------------------------------------------------------

Please enter the amount of the money will you donate monthly:
Please do not use non-digit characters.It can cause some problems.
Money quantity: 200

Please enter which day of each month you would like to donate:
Please enter a number between 1 and 28.
(Otherwise that can cause some problems):
The day: 30
-------------------------------------------------------------
The day you entered could not be accepted.

Please enter a number between 1 - 28: 25
The day you chose has been accepted.

You are directed to the payment page.
Press enter to continue...
```

When you are redirected to the payment page, you will be prompted to enter this information. you will get an output based on this information



```
Please enter your card numbers ([4]-[4]-[4]-[4]): 1111 1111 1111 1111
Please enter expiration date of your card (MM-YY): 06-28
Please enter your CVV code (xxx): 654

Card numbers you have entered: 1111 1111 1111 1111
Expiration Date:  06-28
CVV code: 654
Thanks for your donation.

Press enter to continue...
```

This page is an display of your donation.



```
You will donate 200EUR on 25 of the every month from Germany.
Thanks for your Donation...
```