--Assesment 2--

Problem statement

Travel Agency Management

Specifications:

Variables: Package ID, destination, price, and availability.

Static & Const: Static variable for total packages; const for maximum packages.

Switch Case: Menu for adding, booking, and viewing travel packages.

Looping Statements: Loop through package list.

Pointers: Pointer for dynamic allocation of package details.

Functions: Separate functions for each travel operation.

Arrays: Store package details.

Structures: Structure for travel packages.

Nested Structures: Nested structures for package and customer details.

Unions: Union for payment methods.

Nested Unions: Nested union for different payment options.

Output Expectations: Display travel packages and booking details.

Menu Example:

1. Add Package

2. Book Package

3. View Packages

4. Exit

---solution Code--

```
#include<stdio.h>
#include<stdbool.h>
#include<string.h>

#define MAX_PACKAGES 30
```

```
union payment_details {
    char payment_method;
    int payment_id;
};

struct package {
    int packageid;
    char destination[30];
    float price;
    char availability;
    char booked;
    struct {
        char customer_name[30];
        int customer_age;
        union payment_details det;
    } customer;
};

void add_package(struct package arr[], int size) {
    for(int i = 0; i < size; i++) {
        printf("Enter the details for package %d\n", i + 1);

        printf("Enter the package ID: ");
        scanf("%d", &arr[i].packageid);

        printf("Enter the destination: ");
        getchar();
        fgets(arr[i].destination, 30, stdin);
        arr[i].destination[strcspn(arr[i].destination, "\n")] = '\0';
        printf("Enter the price: ");
        scanf("%f", &arr[i].price);

        printf("Enter the availability ('y' for available, 'n' for not available): ");
        getchar();
        scanf("%c", &arr[i].availability);

        printf("Enter customer name: ");
        getchar();
        fgets(arr[i].customer.customer_name, 30, stdin);
        arr[i].customer.customer_name[strcspn(arr[i].customer.customer_name, "\n")] = '\0';

        printf("Enter customer age: ");
        scanf("%d", &arr[i].customer.customer_age);

        printf("Enter payment method ('c' for card, 'p' for PayPal): ");
        getchar();
        scanf("%c", &arr[i].customer.det.payment_method);

        printf("Enter payment ID: ");
        scanf("%d", &arr[i].customer.det.payment_id);
        arr[i].booked = 'f';
    }
    printf("Successfully registered %d packages!\n", size);
}

void book_package(struct package arr[], int package_id, int count) {
    for(int i = 0; i < count; i++) {
        if(arr[i].packageid == package_id) {
            if(arr[i].booked == 'f') {
                arr[i].booked = 't';
                printf("Package ID %d has been successfully booked!\n", package_id);
            } else {
                printf("Package ID %d is already booked.\n", package_id);
            }
            return;
        }
    }
    printf("Package ID %d not found!\n", package_id);
}
void view_package(struct package arr[], int package_id, int count) {
    for(int i = 0; i < count; i++) {
        if(arr[i].packageid == package_id) {
            printf("\nBooking details for Package ID %d:\n", package_id);
            printf("Destination: %s\n", arr[i].destination);
            printf("Price: %.2f\n", arr[i].price);
            printf("Availability: %c\n", arr[i].availability);
            printf("Booked: %c\n", arr[i].booked == 't' ? "Yes" : "No");
            printf("Customer Name: %s\n", arr[i].customer.customer_name);
            printf("Customer Age: %d\n", arr[i].customer.customer_age);
            printf("Payment Method: %c\n", arr[i].customer.det.payment_method);
            printf("Payment ID: %d\n", arr[i].customer.det.payment_id);
            return;
        }
    }
    printf("Package ID %d not found!\n", package_id);
}
```

```
int main() {
    struct package arr1[MAX_PACKAGES];
    bool is_on = true;
    int count = 0;

    while(is_on) {
        int user_option;
        printf("\n1. Add Package\n2. Book Package\n3. View Package\n4. Exit\nEnter your option: ");
        scanf("%d", &user_option);

        switch(user_option) {
            case 1:
                if (count < MAX_PACKAGES) {
                    int n;
                    printf("Enter the number of packages to add: ");
                    scanf("%d", &n);
                    if(count + n <= MAX_PACKAGES) {
                        add_package(arr1, n);
                        count += n;
                    } else {
                        printf("Cannot add more packages, maximum limit reached.\n");
                    }
                } else {
                    printf("Maximum number of packages reached.\n");
                }
                break;

            case 2:
                int m;
                printf("Enter the package ID to book: ");
                scanf("%d", &m);
                book_package(arr1, m, count);
                break;

            case 3:
                int o;
                printf("Enter the package ID to view: ");
                scanf("%d", &o);
                view_package(arr1, o, count);
                break;

            case 4:
                printf("Exiting...\n");
                is_on = false;
                break;

            default:
                printf("Invalid option! Please try again.\n");
        }
    }

    return 0;
}
```