In this part you are to arrange a variety of Coffees into a subscription package for customers of our Coffee roasting company. The structure describing our Coffee varities is shown below:

```
enum class roastType
{
   light, medium, dark
};


struct Coffee
{
   std::string name;
   roastType roast;
   std::string flavorProfile;
   double pricePerKG{};
};
```

Create a function named `createSubscription` that receives a stock of `Coffee` as the parameter (`std::vector`) and returns a list of `Coffee`. This functions performs the following tasks:

1. Print to the screen the number of coffees in the stock that have a `Creamy` or `Full Bodied` flavor profile

2. Sort the stock of coffee based on their `roastType` (in the order of `light`, `medium`, `dark`)

3. Create a list of `Coffees` (which will be the return value of this function) that contains only `light` and `medium` roasts where the price is less than 10 dollars per KG

4. Add to the subcription of `Coffees` (to be returned) the first dark roast coffee in the stock

5. Tally up the total price per KG of all the `Coffees` in the subscription and print it to the screen

Do not use Manual loops!

Do not use the same algorithm more than once!

Your solution should only contain this function.

Main:

```
int main() {
  std::vector<Coffee> coffees({
    { "Pacific Pipeline", roastType::medium, "Full Bodied", 8.99 },
    { "Three Sisters", roastType::medium, "Tropical Fruit", 7.99 },
    { "Hola", roastType::light, "Juicy", 11.99 },
    { "Horse Power", roastType::dark, "Creamy", 8.99 },
    { "Decaf", roastType::dark, "Creamy", 5.99 },
    { "Grizzly Claw", roastType::light, "Full Bodied", 9.99 },
    });

  // The resulting coffee subscription
  auto coffee_sub = createSubscription(coffees);

  void(*roastFunc[3])() = {
    []() { cout << "Light"; },
    []() { cout << "Medium"; },
    []() { cout << "Dark"; }
  };

  cout << "\nCoffee Subscription" << endl;
  for (auto& x : coffee_sub) {
    cout << x.name << " | ";
    roastFunc[(int)x.roast]();
    cout << " | " << x.flavorProfile << " | " << x.pricePerKG << endl;
  }

  return 0;
}
```