

Metodologie di programmazione (itinerario B, 12 aprile 2022)

1. Mostrare il processo di risoluzione dell'overloading per le seguenti chiamate di funzione. Per ogni chiamata, indicare: (a) l'insieme delle funzioni candidate; (b) l'insieme delle funzioni utilizzabili; e, se esiste, (c) la migliore funzione utilizzabile.

```
namespace N {
    struct T {
        T() = default;
        explicit T(int i);
        T(const std::string& s);
    };
    void fun(int i);           // funzione #1
    void fun(const T& t);      // funzione #2
} // namespace N

void fun(T& t);               // funzione #3
void fun(double d);          // funzione #4

int main() {
    N::T t;
    const N::T& rt = t;
    int i = 1234;
    std::string s = "1234";

    fun(t);                   // chiamata A
    fun(rt);                   // chiamata B
    fun(i);                     // chiamata C
    using N::fun;
    fun(i);                     // chiamata D
    N::fun(s);                 // chiamata E
}
```

2. Individuare e correggere gli errori di progettazione (che sono causa di potenziali scorrettezze e/o inefficienze) nell'interfaccia della classe seguente:

```
class CartaPrepagata {
    using Info = std::string;
    using Money = /* dettaglio implementativo */;
    using Movimento = /* dettaglio implementativo */;
    using Movimenti = std::vector<Movimento>;
    Info codice;
    Money saldo;
    Movimenti movimenti;
public:
    CartaPrepagata(Info codice);
    CartaPrepagata(CartaPrepagata cp);
    CartaPrepagata operator=(CartaPrepagata cp);
    Info getCodice();
    Money get_saldo();
    void ricarica(Money importo);
    void detrai(Money importo);
    void trasferisci_a_carta(Money importo, CartaPrepagata cp);
    Movimenti& movimenti_del_mese();
    std::ostream operator<<(std::ostream out);
};
```

3. La seguente funzione restituisce il valore `true` se e solo se la stringa `s` è palindroma (ovvero, contiene la stessa sequenza di caratteri sia quando è letta da sinistra verso destra, sia quando è letta da destra verso sinistra).

```
bool palindroma(const std::string& s) {
    unsigned i = 0, j = s.size();
    if (i == j)
        return true;
    --j;
    while (i != j) {
        if (s[i] != s[j])
            return false;
        ++i;
        if (i == j)
            return true;
        --j;
    }
    return true;
}
```

Fornire una versione generica della funzione (prototipo e implementazione) che sia in grado di lavorare su una sequenza di elementi di tipo qualunque. Indicare i requisiti sugli iteratori usati per rappresentare la sequenza.

4. Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumendo che le variabili `sc`, `i`, `ul`, `ai` abbiano tipo, rispettivamente, `signed char`, `int`, `unsigned long`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita.

Indice	Argomento	Parametro formale
(a)	<code>sc</code>	<code>int</code>
(b)	<code>(ul - i)</code>	<code>unsigned long</code>
(c)	<code>5.5</code>	<code>float</code>
(d)	<code>i</code>	<code>const int&</code>
(e)	<code>ai</code>	<code>int*</code>
(f)	<code>ai[5]</code>	<code>const int*</code>

5. Fornire un semplice esempio di violazione della ODR (regola della definizione unica) che possa essere facilmente rilevato dal compilatore (in senso stretto, cioè prima della fase di collegamento, affidata al linker).
6. Pur assumendo la correttezza delle funzioni `apri_file`, `chiudi_file` e `esegui_query`, il seguente codice potrebbe non avere un comportamento corretto in presenza di eccezioni:

```
void work(DB_Connection& db_conn,
         const std::vector<std::string>& queries) {
    File* fres = apri_file("db_results.out");
    File* flog = apri_file("db_log.out");
    for (auto i = 0; i < queries.size(); ++i)
        esegui_query(db_conn, queries[i], fres, flog);
    chiudi_file(flog);
    chiudi_file(fres);
}
```

Descrivere il problema e fornire una soluzione basata sull'uso dell'idioma RAII (“*l’acquisizione di risorse è una inizializzazione*”).